

POLITECHNIKA SZCZECIŃSKA

Instytut Inżynierii Chemicznej i Chemii Fizycznej

Irena Kuźniewska-Jach

**OBLICZENIA PROJEKTOWE PÓŁEK
APARATÓW KOLUMNOWYCH**

SZCZECIN 1991

0471
POLITECHNIKA SZCZECIŃSKA

Instytut Inżynierii Chemicznej i Chemii Fizycznej

Irena Kuźniewska-Lach

**OBLICZENIA PROJEKTOWE PÓLEK
APARATÓW KOLUMNOWYCH**

SZCZECIN 1991

66.021.3 : 66.01 : 649.688(075.8)
PI 959



CH-6573

Recenzent

Andrzej Kałed

Dpracowanie redakcyjne

Jolanta Orkisz

Wydano za zgodą

Rektora Politechniki Szczecińskiej

z dnia 2.08.1991

Biblioteka Główna
Politechniki Szczecińskiej

CH II - 6573



004-006573-00-0

PI 959

Wyd. I. Nakład 200+23 egz. Ark. wyd. 10,5. Ark. druk. 16,9

Papier offsetowy kl. III 71 g B₁ Oddano do druku 2.08.91

Druk ukończono w grudniu Zamówienie nr 281/91

Cena zł. 7350

Zakład Poligraficzny Politechniki Szczecińskiej "ZAPOL"

71-065 Szczecin, al. Piastów 42a

SPIS TREŚCI

WSTĘP	7
1. STRUKTURA PROGRAMÓW	11
2. WPROWADZANIE I WSTĘPNE PRZETWARZANIE DANYCH	13
3. WYPROWADZANIE WYNIKÓW	17
3.1. Analiza wyników	18
4. OPIS PROGRAMÓW	21
4.1. Program KOLP	22
4.2. Program ZAWOR	24
4.3. Program BITD	24
4.4. Programy BITBEZ i RUBZT	25
5. SCHEMATY BLOKOWE PROGRAMÓW	27
5.1. Schemat blokowy programu KOLP	28
5.2. Schemat blokowy programu ZAWOR	41
5.3. Schemat blokowy programu BITD	59
5.4. Schemat blokowy programu BITBEZ	70
5.5. Schemat blokowy programu RUBZT	78
6. TABULOGRAMY PROGRAMÓW	85
6.1. Tabulogram programu KOLP	86
6.2. Tabulogram programu ZAWOR	104
6.3. Tabulogram programu BITD	123
6.4. Tabulogram programu BITBEZ	137
6.5. Tabulogram programu RUBZT	148
7. WYKAZ PROCEDUR I FUNKCJI	159
8. SCHEMATY BLOKOWE PROCEDUR	161
9. TABULOGRAMY PROCEDUR	197
10. PRZYKŁAD OBLICZEŃ	227
10.1. Dane	227
10.2. Wyniki obliczeń	230
11. WYKAZ OZNACZEŃ	239
12. WYKAZ LITERATURY	251

Skrypt przeznaczony jest dla studentów IV roku kierunku "Inżynieria chemiczna" wykonujących projekty w ramach przedmiotu "Optymalizacja procesowa". Może być też pomocą przy rozwiązywaniu zagadnień projektowych z przedmiotów: "Inżynieria chemiczna" (wymiana masy) oraz "Technologia i inżynieria systemów".

Skrypt zawiera algorytmy obliczeń konstrukcyjnych aparatów kolumnowych wyposażonych w następujące typy półek: kołpakowe, zaworkowe V1 i V4, sitowe z przelewem oraz sitowe i rusztowe bezprzelewowe. Opracowane programy, obok obliczeń wymiarów konstrukcyjnych zgodnych z obowiązującymi normami, pozwalają na wybór optymalnego rozwiązania konstrukcyjnego w ramach określonego typu półki. Szczegółowe schematy blokowe oraz załączone tabulogramy programów powinny ułatwić studentom śledzenie toku obliczeń i przygotowanie zbiorów danych do przeprowadzenia obliczeń dla dowolnego układu.

Pragnę podziękować w tym miejscu mgr inż. Sylwestrowi Tutko za pomoc przy graficznym opracowaniu schematów blokowych.

WSTĘP

Projektując kolumnę, czyli ustalając jej wymiary, należy mieć na uwadze następujące aspekty:

- 1) spełnienie założeń procesowych,
- 2) zapewnienie najkorzystniejszych warunków pracy,
- 3) minimalizację kosztów inwestycyjnych i eksploatacyjnych.

Rozpatrzmy przypadek projektu kolumny rektyfikacyjnej. Przystępując do obliczeń dysponujemy najczęściej danymi odnośnie składu i natężenia przepływu surowki oraz składu i natężenia odbioru produktu (np. destylatu). Pierwszym krokiem, jaki należy wówczas wykonać jest sporządzenie bilansu ogólnego i obliczenie średnich natężeń przepływu cieczy i par w kolumnie. Do sporządzenia bilansu masowego kolumny rektyfikacyjnej niezbędna jest znajomość stopnia deflegacji. W literaturze [33] podano zarówno metody obliczania minimalnej ($R_{\min}$) jak i optymalnej (R_{opt}) wartości stopnia deflegacji. Optymalny stopień deflegacji związany jest, jak wiadomo [12] z minimum sumy kosztów, na które składają się koszty budowy samej kolumny (K_K), koszty urządzeń wymiany ciepła (K_W) i koszty energetyczne (K_E). Wyznaczając optymalną wartość stopnia deflegacji, określamy więc najkorzystniejsze warunki pracy kolumny. Obliczenia projektowe półek wykazały, że warunki te mogą być spełnione przez wiele wariantów konstrukcyjnych przyjętego typu półki. Poszczególne rozwiązania mogą się różnić np.: odległością między półkami, wysokością i usytuowaniem przegrody przelewowej, liczbą i rozmieszczeniem elementów rozpraszających fazę gazową na półce itp. Różnice te mogą przy tym dotyczyć jednej lub kilku cech konstrukcyjnych. Zastosowanie maszyn cyfrowych w procesie projektowania pozwala na bardzo szybkie przeliczenie wszystkich możliwych wariantów konstrukcyjnych. Nasuwa się pytanie, które z otrzymanych rozwiązań stabilnie pracującej półki jest najlepsze, a więc na

jakiej podstawie należy dokonać wyboru jednego z wielu rozwiązań?
Zwykle kryterium wyboru wiąże się z kosztami.

Wg Colburna [12] koszty kolumny mogą być wyrażone w postaci zależności:

$$K_k = f \left(\frac{D \cdot n}{E \cdot h} \right), \quad (1)$$

gdzie: D - średnica kolumny,

n - liczba półek teoretycznych,

E - sprawność półki,

h - czas pracy użytecznej.

W koszcie kolumny, kosztowi każdej półki można przyporządkować koszt odpowiedniej części płaszcza kolumny (zawartego między półkami). Taki zespół określać będziemy dalej mianem segmentu kolumny, a jego koszt kosztem segmentu (K_{sk}). Dla danego typu półki i ustalonego materiału konstrukcyjnego koszt segmentu można wyrazić zależnością:

$$K_{sk} = f(D, H), \quad (2)$$

gdzie: H - odległość między półkami.

Przyjmując czas pracy użytecznej za wartość stałą i uwzględniając, że dla ustalonego stopnia deflegacji liczba półek teoretycznych jest określona, można stwierdzić, że koszt kolumny jest wprost proporcjonalny do ilorazu:

$$K_k = K_{sk} / E. \quad (3)$$

Zdarza się jednak, że z punktu widzenia procesu istotne są też inne czynniki, które poza kosztami i sprawnością, powinny być uwzględnione przy wyborze optymalnego aparatu. Np. w przypadku absorpcji lub rektyfikacji próżniowej najodpowiedniejsza będzie półka charakteryzująca się możliwie małym spadkiem ciśnienia, a dla procesu rektyfikacji z reakcją chemiczną w fazie ciekłej, półka, która zapewnia

1 STRUKTURA PROGRAMÓW

Każdy z przedstawionych programów opatrzony jest nazwą, kojarzącą się z typem półki, którą za pomocą tego programu można zaprojektować. Przyjęto następujące nazwy:

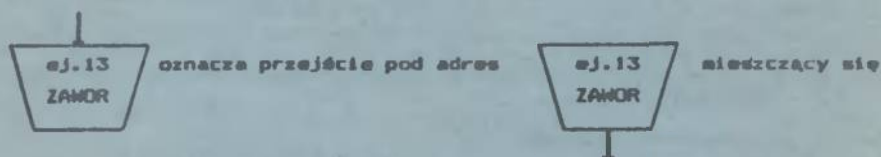
- KOLP - dla półek kołpakowych,
- ZAMOR - dla półek zaworkowych z zaworkami typu VI i V4,
- SITO - dla półek sitowych z przelewem,
- SITBEZ - dla półek sitowych bezprzelewowych,
- RUSZT - dla półek rusztowych.

W każdym z programów można wyróżnić trzy bloki:

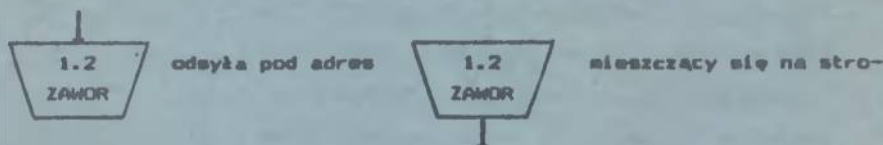
- 1) wprowadzanie i przetwarzanie danych,
- 2) część obliczeniowa,
- 3) wyprowadzanie wyników.

Algorytmy bloków 1 i 3 są we wszystkich programach identyczne. Zasadnicze różnice w poszczególnych programach występują między blokami 2, na które składają się równania i korelacje służące do obliczeń konstrukcyjnych i hydrodynamicznych określonego typu półki. Algorytmy, według których zaprogramowano obliczenia, przedstawiono w formie schematów blokowych. Z uwagi na dydaktyczny charakter niniejszego opracowania schematy te są bardzo szczegółowe. Procedury opatrzone wykazem parametrów (o ile takie istnieją) przekazywanych bądź to przez zmienną, bądź przez wartość oraz wykazem zmiennych lokalnych. Zmiennie na schematach blokowych oznaczono zgodnie z wymogami normy [20], lub też ze zwyczajowo stosowanymi [33] oznaczeniami. Natomiast nazwy zmiennych w programach mogą przyjmować jedynie ściśle zdefiniowaną w danym języku programowania postać. Stąd np. gęstość cieczy na schematach oznaczono symbolem " ρ_c " natomiast w programach wielkość tę reprezentuje zmienna "roc". Z tego względu zmiennie w wykazie oznaczeń wyspecyfikowano w dwóch kolumnach. Poniżej przy tym

opis zmiennych pomocniczych. Schematy blokowe wszystkich programów, a także niektórych procedur są dość rozbudowane. Często istnieje więc konieczność wykonywania "skoków" do odległych bloków. Wymaga to stosowania tzw. łączników określających jednoznacznie adres, pod który jest kierowany dalszy tok obliczeń. Rozróżnienie łączników międzystronicowych - mających kształt trapezu, od łączników wewnątrz danej strony (okręgi) powinno ułatwić śledzenie algorytmu. Każdy łącznik zawiera charakterystyczny dla niego identyfikator, numer strony danego schematu oraz nazwę procedury lub programu, wewnątrz których obowiązuje. Identyfikatorem jest liczba jednocyfrowa lub (w przypadku skoku warunkowego) ciąg liter bądź cyfr, zgodny z etykietą występującą w danym programie. Dla przykładu łącznik :



na stronie 13 schematu blokowego programu ZAWOR. W programie ZAWOR występuje etykieta "ej". Natomiast łączniki:



nie 2 schematu blokowego programu ZAWOR. Nie jest to skok warunkowy, a jedynie przejście na nową stronę schematu blokowego. Etykiety "1" w programie ZAWOR nie są. W przypadku schematów blokowych procedur mniej rozbudowanych zaniechano podawanie w łącznikach numerów stron.

2. WPROWADZANIE I WSTĘPNE PRZETWARZANIE DANYCH

Przeprowadzenie obliczeń konstrukcyjnych i sprawdzenie warunków hydrodynamicznych na półkach jest możliwe, jeśli znane są następujące wielkości:

- 1) natężenia przepływu strumieni oraz właściwości fizykochemiczne mieszanin cieczy i par w tej części kolumny, dla której mają być wykonane obliczenia,
- 2) ograniczenia konstrukcyjne narzucone przez odpowiednie normy branżowe,
- 3) dane wynikające z empirycznego charakteru korelacji, wymagającego niekiedy korzystania z wykresów, tablic czy nomogramów,
- 4) przyjęte, jako zadane w rozważanym układzie wielkości takie, jak: współczynnik asocjacji, współczynnik pienienia itp.

Dane ujęte w punktach 2, 3 i 4, charakterystyczne dla rozważanego układu i wybranego typu konstrukcyjnego półki, wprowadzane są na początku każdego programu. Na schematach blokowych, odpowiadające tym wielkościom zmiennie wyspecyfikowano w blokach CZYTANIE. Po uruchomieniu programu wartości tych zmiennych przekazywane są ze zbiorów o nazwach kojarzących się z nazwami poszczególnych programów. Nazwy zbiorów danych zaczynają się trzyliterowym ciągiem "DAN" lub "TAB", po którym następuje pełna lub skrócona (z przyczyn formalnych) nazwa programu. Tak np. w zbiorach DANKOLP i TABKOLP zestawiono dane czytane po uruchomieniu programu KOLP. Kontynuacja obliczeń możliwa jest po ustaleniu wartości natężeń przepływu i właściwości fizykochemicznych strumieni cieczy i par. Jeżeli wartości tych zmiennych są znane, należy je wprowadzić do zbioru DANE. W przeciwnym przypadku do wyboru są następujące możliwości:

- 1) Obliczenia dotyczą półki aparatu kolumnowego o dowolnym przeznaczeniu. Przygotowujemy dane odnośnie natężeń przepływu oraz temper-

ratur i składów strumieni cieczy i par w skrajnych przekrojach kolumny. Wartości te wprowadzamy do zbioru DANE2.

2) Obliczenia dotyczą kolumny rektyfikacyjnej, dla której sporządzono wstępny bilans masowy. Jako dane do obliczeń przyjmujemy wówczas: natężenie przepływu, skład i temperaturę surowki oraz natężenia przepływu i stężenia składników w cieczy wyczerpanej i w destylacie. Wartości te wprowadzić należy do zbioru DANE3.

3) W przypadku obliczeń półki kolumny rektyfikacyjnej, której bilans masowy nie jest znany, przygotowujemy jeden z następujących zestawów danych:

- a) natężenia przepływów i udziały składników w surowce i w destylacie,
- b) natężenia przepływów i udziały składników w surowce i w cieczy wyczerpanej,
- c) natężenie przepływu surowki oraz udziały składników w surowce, w destylacie i w cieczy wyczerpanej.

Odpowiednie wartości należy wprowadzić do zbioru DANE1, a po uruchomieniu programu wybrać opcję (a, b lub c) zgodną z wprowadzonymi danymi.

Odsyłanie do poszczególnych zbiorów i kontakt z użytkownikiem odbywa się za pośrednictwem procedury FIZCHEM (str. 164).

W przypadku obliczania kolumny rektyfikacyjnej sporządzany jest bilans masowy procesu (procedura BILANS - str. 162) przy założeniu, że surowka jest cieczą wrzącą. Wartość stopnia deflegacji podaje się z klawiatury. Temperaturę wrzenia cieczy o określonym składzie oraz stężenia składników w fazie gazowej, pozostającej w równowadze z tą cieczą, oblicza się za pomocą procedury TEMP (str. 181). Do obliczania stałych równowag składników zastosowano metodę Wilsona [31] (procedura KRO - str. 177). Właściwości fizykochemiczne składników, takie jak: masy cząsteczkowe, parametry krytyczne, wartości reochory i parachory, stałe Sutherlanda, tablice gęstości w różnych temperaturach itp., zestawione w zbiorze TABWAR. Jako źródło tych danych może służyć np. "Poradnik Fizykochemiczny" [28].

Zbiory DANE, DANE1, DANE2 oraz TABWAR są zbiorami wspólnymi dla

wszystkich programów. Jeżeli stężenia składników w zbiorach DANE1 lub DANE2 wyrażone są w ułamkach wagowych, to natężenia przepływu strumieni należy podać w kg/h. Wprowadzenie danych w takiej postaci powoduje uruchomienie procedur ULNAG (str. 186) i PRZELICZ (str. 167). Następuje przeliczenie ułamków wagowych na molowe oraz natężeń przepływu strumieni z kg/h na kmol/h. Znajomość temperatury strumieni i stężeń składników w obu fazach pozwala na przejście do procedury MLAS (str. 188), służącej do obliczania średnich właściwości fizykochemicznych mieszanin cieczy i par.

Do procedury MLAS włączono funkcje:

- MAC (str. 170) - do obliczania masy cząsteczkowej cieczy i par oraz gęstości par,
- ROR (str. 167) - do obliczania gęstości cieczy,
- SIR (str. 174) - do obliczania napięcia powierzchniowego,
- TER (str. 174) - do obliczania lepkości cieczy,
- TEIA (str. 175) - do obliczania lepkości par.

Algorytmy tych funkcji ustalono w oparciu o równania i korelacje podane w literaturze [5]. W przypadku obliczeń prowadzonych dla kolumny rektyfikacyjnej - ednie parametry fizykochemiczne obliczane są za pomocą procedury MLASFIZ (str. 189). Wywołanie tej procedury powoduje dwukrotne uruchomienie procedur TEMP i MLAS. Raz w celu wykonania odpowiednich obliczeń dla zasilania i powtórnie dla destylatu lub cieczy wyczerpanej. Ostatnie działanie w ramach procedury MLASFIZ sprowadza się do obliczenia średnich arytmetycznych z wartości otrzymanych dla zasilania i destylatu, jeśli projektowana będzie półka dla kolumny górnej lub z wartości otrzymanych dla zasilania i cieczy wyczerpanej, gdy obliczenia projektowe dotyczyć mają dolnej części kolumny.

3. WYPROWADZANE WYNIKÓW

Wyniki obliczeń projektowych wyprowadzane są do zbiorów o nazwach skojarzonych z nazwą określonego programu. Nazwy te rozpoczynają się trzyliterowym ciągiem " WYN ", po którym następuje pełna lub skrócona nazwa programu. Tak np. wyniki programu ZAWOR wyprowadzane są do zbioru WYNZAWOR. Do każdego zbioru wyprowadzany jest opis zawierający nazwę typu konstrukcyjnego półki oraz wartości natężeń przepływu fazy i stopnia deflegacji, dla których prowadzone są obliczenia. W trakcie obliczeń do zbiorów wynikowych wyprowadzane są zestawy wszystkich możliwych, dla danego typu półki, rozwiązań konstrukcyjnych. Każdy taki zestaw zawiera nazwy zmiennych i jednostki, w jakich wyrażono poszczególne wielkości. Prócz wymiarów szczegółów konstrukcyjnych, do omawianych zbiorów wyprowadza się też aktualnie obliczone:

- wielkości hydrodynamiczne (jak spadek ciśnienia, porywanie itp.) istotne z punktu widzenia zarówno hydrauliki jak i sprawności projektowanej półki,
- koszt segmentu kolumny, liczony jako suma kosztu półki i kosztu płaszcza między półkami,
- sprawność kolumny,
- wartości kryterium wyboru.

Wszystkie zmienne wyprowadzane jako wyniki obliczeń wyspecyfikowano na schematach w blokach WYNIKI.

W przypadku, gdy do wstępnie przyjętych założeń nie można dobrać pozostałych wymiarów konstrukcyjnych zapewniających prawidłową pracę półki, do zbioru wyprowadzany jest komentarz informujący o rodzaju lub przyczynie zaburzeń pracy. Zbiory wynikowe zakładane są na dysku twardym w katalogu o nazwie POLKI. Nie przewiduje się programowego wyprowadzania wyników na drukarkę. Takie rozwiązanie skraca czas działania programów i daje możliwość zaoszczędzenia papieru. Przeglą-



dając zbiór można dokonać analizy wyników i na jej podstawie podjąć decyzję o wydruku całości, bądź tylko wybranych fragmentów zbioru. Zbiory wyników zachowywane są do ponownego uruchomienia programu.

3.1. Analiza wyników

Po zakończeniu obliczeń konstrukcyjnych półki może się okazać, że:

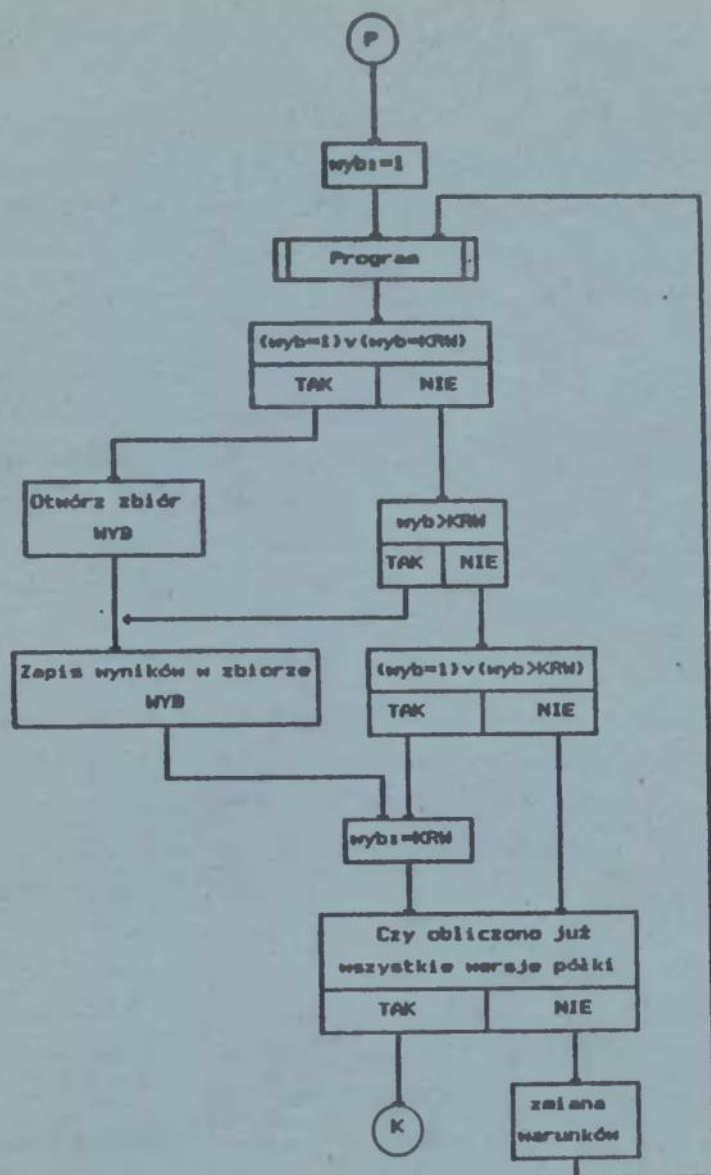
- 1) zadane parametry ruchowe leżą poza obszarem stabilnej pracy tego typu półki - brak rozwiązania,
- 2) istnieje tylko jedno rozwiązanie konstrukcyjne zapewniające spełnienie ograniczeń normowych i warunków hydrodynamicznych,
- 3) otrzymano więcej niż jedno rozwiązanie.

Przypadek 1 eliminuje dany typ półki, jako nieprzydatny w rozważanym układzie i skłania do szukania rozwiązań dla innych typów konstrukcyjnych. Przypadek 2 nie wymaga komentarza. Natomiast w przypadku 3 należy dokonać wyboru jednego z wielu rozwiązań. Podjęcie decyzji ułatwiają określone na wstępie kryteria wyboru (równania 4 - 7). Aby umożliwić korzystanie z proponowanych kryteriów do programów wprowadzono zależności matematyczne do obliczania kosztów segmentu kolumny, sprawności kolumny oraz objętości cieczy zatrzymanej na półce.

Koszt segmentu, liczony jako suma kosztu półki i kosztu płaszczyzny między półkami, wyznacza się na podstawie dostępnych w literaturze [3] równań.

Sprawność kolumny z półkami przelewowymi obliczana jest w oparciu o sprawność umowną półki [10,34]. Obliczenia wykonywane są w ramach procedury SPRAMNOSC, której schemat blokowy przedstawiono na str.179. Natomiast przy obliczaniu sprawności kolumny z półkami bez elementów wpływowych, zastosowano określony w literaturze [18,34], tok postępowania, bazujący na liczbie jednostek przenikania masy. Schemat blokowy procedury SPRAMBEZ przedstawiono na str. 196.

O przyjęciu kryterium, według którego będzie wybierane rozwiązanie optymalne, decyduje użytkownik po uruchomieniu programu. Istnieje możliwość ustalenia zupełnie innego kryterium niż zaproponowane.



Algorytm wyboru i przekazywania rozwiązań optymalnych do zbiorów wynikowych "MYB".

Wymaga to dokonania odpowiednich zmian w zbiorze KRYTYWYB (patrz rozdział 4 - Opis programów).

Przegląd zbiorów wynikowych, w celu analizy wyników i wyboru rozwiązania optymalnego, jest czasochłonny. Aby tę czynność pośmagnąć, w każdym z programów otwarto dodatkowe zbiory, do których wyprowadzane są wyniki charakteryzujące się minimalną wartością przyjętego kryterium wyboru. Nazwy tych zbiorów zaczynają się trzyliterowym ciągiem "WYB", po którym następuje pełna lub skrócona nazwa programu. Tak np. w zbiorze WYBIAWOR należy szukać optymalnego rozwiązania pólki zaworkowej. Jeżeli zbiór ten okaże się zbiorem pustym, co świadczy o braku rozwiązania, można odwołać się do zbioru WYVIAWOR, w którym zestawione będą komentarze informujące o przyczynie takiej sytuacji. W przypadku istnienia tylko jednego rozwiązania, wyniki obliczeń będą również zapisane w zbiorze WYBIAWOR. Zbiory WYB zakładane są na dysku twardym w katalogu POLKI i zachowywane do ponownego uruchomienia programu.

Algorytm wyboru i przekazywania rozwiązania optymalnego do odpowiedniego zbioru przedstawiono na schemacie blokowym (str. 19).

4. OPIS PROGRAMÓW

Obliczenia konstrukcyjne pólek zaprogramowano w języku Turbo Pascal 4.0 na mikrokomputer klasy IBM. Programy te zapisane są na dyskietce ⁸⁾, w katalogu POLKI w zbiorach, których nazwy zgodne są z nazwami programów. Utworzono po dwa zbiory o tych samych nazwach. Jedne z nich mają standardowe rozszerzenie .PAS, umożliwiające edycję tekstu na ekranie monitora. Drugie to zbiory z rozszerzeniem .EXE, utworzone w celu umożliwienia wykonywania obliczeń bezpośrednio z poziomu systemu operacyjnego DOS, bez konieczności uruchamiania systemu Turbo Pascal.

Większość procedur i funkcji zapisano w oddzielnych zbiorach łączonych z poszczególnymi programami za pomocą parametrycznej dyrektywy systemowej postaci: (@1 nazwa zbioru). Dotyczy to tych funkcji i procedur, które występują w więcej niż jednym programie. Zbiory te, o nazwach zbliżonych do nazw odpowiednich funkcji lub procedur (wykaz str. 159), zestawione są w podkatalogu PODPROGR.

Na dyskietce (podkatalog DANE) założono też, omawiane w punkcie 2 zbiory danych o nazwach indywidualnych dla każdego programu, a także zbiory: DANE, DANE1, DANE2, DANE3 i TABMAR - wspólne dla wszystkich programów. W tych ostatnich zestawione są pliki danych dla układu będącego podstawą testowania programu (kwas octowy - alkohol etylowy - octan etylu - woda). Przed przystąpieniem do obliczeń dla innego układu dane muszą zostać uaktualnione. Kolejność wprowadzania danych musi być zgodna z kolejnością czytania zmiennych w programach (patrz tabulograpy). Po uruchomieniu dowolnego programu na ekranie monitora będą pojawiały się komentarze informacyjne i pytania. Z komentarzami

⁸⁾ Dyskietka z programami jest dostępna w Instytucie Inżynierii Chemicznej i Chemii Fizycznej Politechniki Szczecińskiej.

należy się zapoznać, a na pytania odpowiadać zgodnie z zaleceniami. Jedno z przykładowych pytań ma postać: "Czy dane są natężenia przepływu i średnie właściwości fizykochemiczne strumieni w kolumnie? t/n". Jeżeli dysponujemy odpowiednimi danymi, to odpowiadamy twierdząco ("t") naciskając klawisz z literą "t". Na monitorze pojawi się wówczas komunikat o treści: "Program korzysta z wartości zestawionych w zbiorze DANE". Oznacza to, że ten właśnie zbiór powinien zostać uaktualniony przed uruchomieniem programu. Po wczytaniu danych na ekranie monitora pojawi się polecenie wskazania kryterium, według którego wybierane będzie rozwiązanie optymalne. Jeśli nie decydujemy się na żadne z proponowanych kryteriów i zamierzamy ustalić własne (opcja 5), nastąpi zatrzymanie programu. Należy wówczas odnaleźć zbiór KRYTWYB (katalog POLKI, podkatalog PODPROGR), pod zieloną "kryt" podstawić ustalone przez siebie wyrażenie i ponownie uruchomić program.

W trakcie obliczeń, na ekranie monitora, może też pojawić się komunikat sygnalizujący błąd w danych. Działanie programu zostanie przerwane. Przed ponownym uruchomieniem programu dane muszą zostać zweryfikowane. Możliwości wystąpienia innych komentarzy omówione są przy opisie poszczególnych programów.

4.1. Program KOLP

Program o nazwie KOLP, napisany zgodnie z algorytmem przedstawionym na schemacie blokowym (str. 28), służy do obliczania pólsek kołpakowych. W programie wykorzystano korelacje empiryczne i równania oraz wykresy i nomogramy podawane w pracach [1,8,19,31,34]. Obliczenia prowadzone są dla wszystkich wartości H , które jako dane [29] wprowadzone są do tablicy $h1$ [1..ih]. Wymiary kołpaka i rurki gazowej (zgodne z normą [25]) ustalane są w czasie działania programu, w zależności od obliczonej średnicy kolumny. Długość krawędzi przelewowej i podziałkę kołpaków, a także ilość kołpaków na półce i sposób ich rozmieszczenia przyjmowane są wg normy [22]. Przy ustalonym H obli-

czenia wykonywane są dla wszystkich wartości h_p czytanych z tablicy h_{p1} [1..3, 1..k]. Pierwszy wskaźnik określany jest przy wyborze wysierów kołpaka, drugi natomiast można zmienić od 1 do k (gdzie k jest ilością różnych wartości h_p , dla których wykonywane będą obliczenia). Wartości h_{pk} wybierane są z tablicy h_{k1} [1..3, 1..k]. Pierwszy wskaźnik określany jest tak samo, jak w przypadku tablicy h_{p1} , natomiast zmianę wskaźnika drugiego zaprogramowano tak, aby można było regulować wartość h_{wp} . Zmiana tej wielkości może wpłynąć na poprawę warunków hydrodynamicznych na półce [34]. Podobny efekt można osiągnąć korygując wstępnie przyjęte wartości l_p i t . Do tego celu służy w programie procedura PRZEL (str. 176), w ramach której następuje zwiększanie stosunku l_p/D i zmiana podziałki kołpaków. W przypadku, gdy skorygowanie wartości tych zmiennych jest niemożliwe (ograniczenia cytowanej normy), do zbioru WYKOLP wyprowadzany jest komentarz informujący o przyczynie niewłaściwej pracy półki przy ustalonych aktualnie wartościach H i h_p . Obliczenia powtarzane są dla kolejnych wartości z tablic h_1 i h_{p1} .

Osobnego omówienia wymaga procedura GRADCIS, służąca do obliczania spiętrzenia cieczy odgazowanej na półce. Algorytm obliczeń, ustalony wg Aleksandrowa [1], zawiera równanie nieliniowe, rozwiązywane w programie metodą Newtona. Jako pierwsze przybliżenie przyjęto wartość $\Delta_0 = 0.0125$. Jeżeli w dwudziestu iteracjach nie otrzyma się rozwiązania z dokładnością równą 0.0005, punkt startowy zostaje zmieniony. Gdy mimo to zbieżność nie zostanie osiągnięta, obliczenia zostaną przerwane. Na monitorze pojawi się komentarz "ALGORYTM NEWTONA ROZBIEŻNY". Przypadek taki wymaga osobnego rozpatrzenia. Podczas testowania programu nie spotkano się z taką sytuacją. W ramach omawianej procedury, wyznaczany jest też współczynnik poprawkowy [8], pozwalający na lepsze przybliżenie wartości Δ . Współczynnik ten obliczany jest w oparciu o dane zestawione w tablicach fg , $b3$ i $c3$. W przypadku, gdy obliczony aktualnie argument funkcji poprawkowej znajdzie się poza przedziałem interpolacji, na ekranie monitora pojawi się odpowiedni komunikat. Program nie zostaje zawieszony.

Obliczenia są kontynuowane dla wartości Δ obliczonej bez uwzględnienia poprawki.

4.2. Program ZAMOR

Program o nazwie ZAMOR służy do obliczania półek zaworkowych - zarówno z zaworkami VI jak i V4. Schemat blokowy obliczeń przedstawiono na str. 41. Algorytm ustalono w oparciu o równania i korelacje empiryczne podane w literaturze [4,15,19,20,34]. W pierwszej kolejności ustalana jest przybliżona długość drogi na półce. W celu kontynuacji obliczeń należy założyć odległość między półkami. Przewidziano wykonanie obliczeń dla każdej z wartości H wczytanych jako dane [29] do tablicy h_i [1..ih]. W trakcie obliczeń zakładana jest też wysokość przegrody przelewowej. Program korzysta ze wszystkich wartości h_p zestawionych w tablicy h_{pi} [1..k]. Wymiary i masę zaworków przyjmuje się zgodnie z normą [26], natomiast podziałkę i liczbę zaworków na półce oraz długość przegrody przelewowej wg norm [24] lub [28]. Jeżeli przy ustalonych parametrach konstrukcyjnych nie zostaną spełnione warunki zapewniające prawidłową pracę półki, korygowana jest ilość lub masa zaworków na półce. Jest to realizowane przez zmianę (w granicach dopuszczalnych przez cytowane normy) wartości l_p i t , lub zastąpienia części zaworków lżejszych cięższymi [26]. W przypadku, gdy zmiany te nie dają pożądanych rezultatów, do zbioru WYNZAMOR wyprowadzany jest komentarz informujący o przyczynie niewłaściwej pracy półki przy obranych wartościach H i h_p . Obliczenia są powtarzane dla kolejnych wartości z tablic h_i i h_{pi} .

4.3. Program SITO

Program ten, służący do obliczania półek sitowych z przelewem, napisano w oparciu o algorytm przedstawiony na schemacie blokowym - str. 39. Równania i korelacje zaczerpnięto z literatury [7,15,16,34]. Obliczenia wykonywane są dla wszystkich wartości H , h_p i d_o , które

jako dane zapisano odpowiednio w tablicach $h1[1..ih]$, $hpl[1..k]$ oraz $dol[1..ido]$. Tablicę hpl ustalono w oparciu o zalecenia literaturowe [34], natomiast wartości w tablicach $h1$ i dol zgodne są z odpowiednimi normami [29, 23]. W trakcie obliczeń korygowane są, w razie potrzeby, wartości $1/p$ oraz wartości t/d_o . Zmiany te dokonywane mogą być jedynie w granicach dopuszczanych normą [23]. W przypadku, gdy przy ustalonych parametrach konstrukcyjnych, nie zostaną spełnione warunki zapewniające prawidłową pracę półki, do zbioru WYNSITO wyprowadzany jest komentarz informujący o przyczynie ziej pracy półki przy obranych wartościach H , h_p i d_o . Obliczenia powtarzane są dla kolejnych wartości z tablic $h1$, hpl i dol .

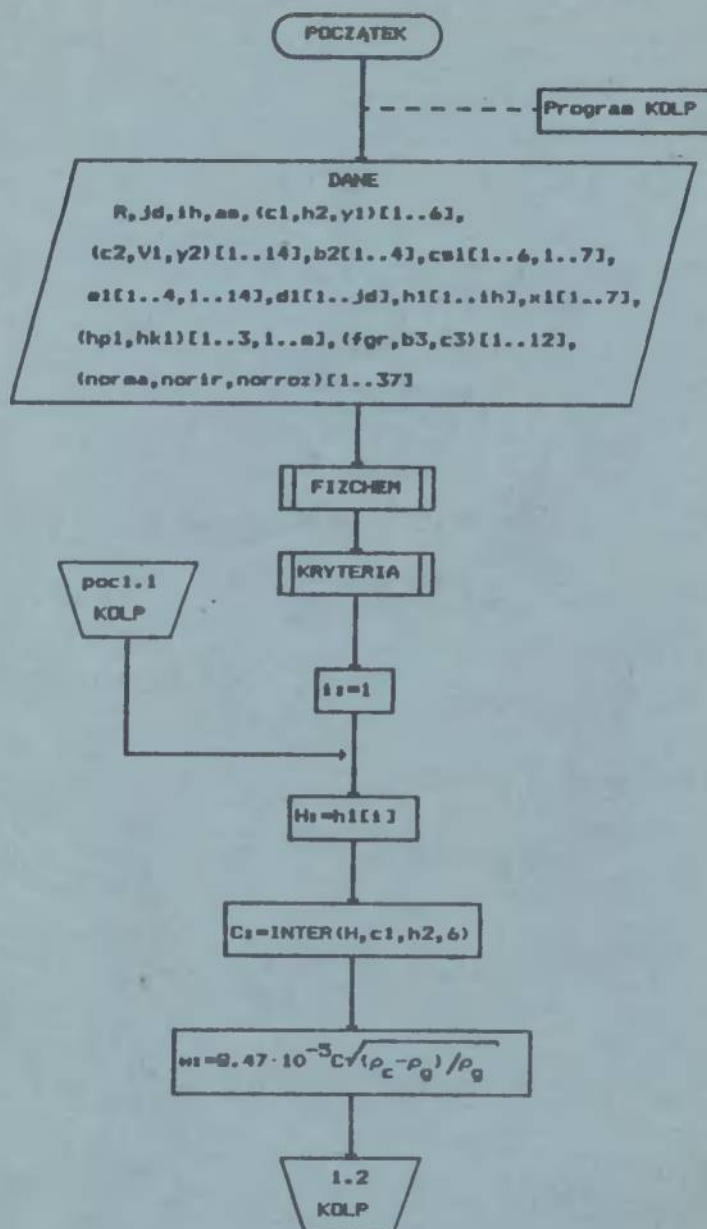
4.4. Programy SITBEZ i RUSZT

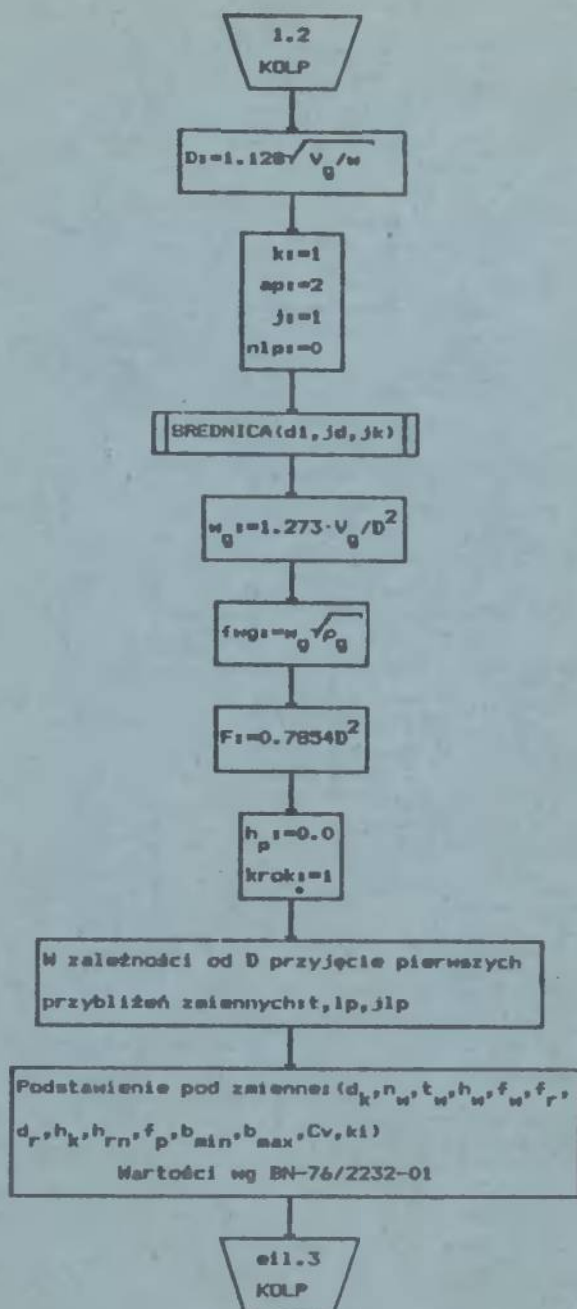
Obydwa programy służą do obliczania półek bezprzelewowych (SITBEZ - półki sitowe, RUSZT - półki rusztowe). Obliczenia wymiarów i hydrauliki tych półek oparto na zależnościach przedstawionych w literaturze [1,2,13,14,32,34]. W obu programach obliczenia wykonywane są dla wszystkich wartości d_z i ϕ zestawionych odpowiednio w tablicach $dol[1..ib]$ oraz $fil[1..ifil]$. Zmianę warunków hydrodynamicznych na każdej z półek można uzyskać poprzez dopuszczalne kombinacje między tymi wielkościami.

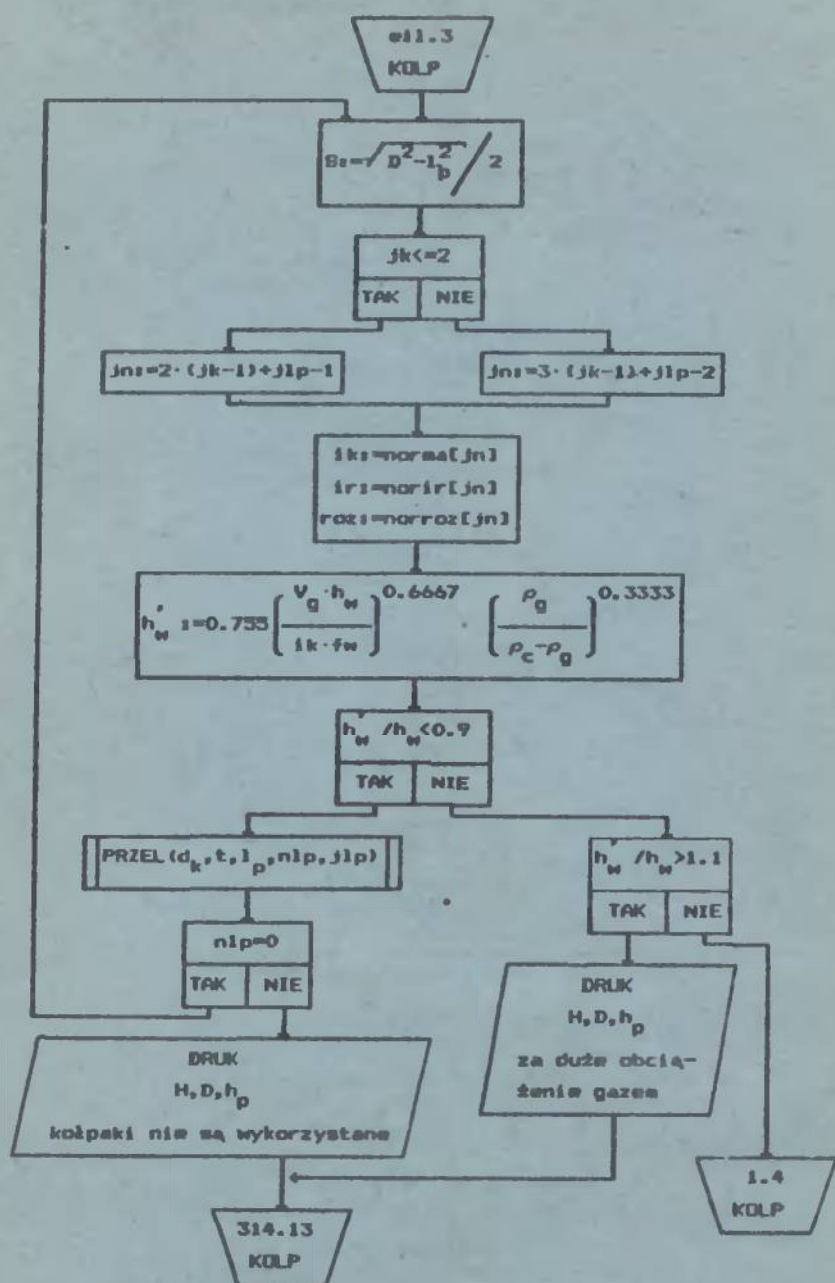
Algorytmy obu programów różnią się. Wymiary półek sitowych są znormalizowane [23], podczas gdy dla półek rusztowych nie opracowano dotychczas polskiej normy. W związku z tym przy doborze średnicy i podziałki otworków kierowano się ograniczeniami cytowanej normy, podczas gdy szerokość i podziałkę szczelin przyjmuje się wg zaleceń podawanych w literaturze [6,9,15,16,34]. Schemat blokowy programu SITBEZ przedstawiono na str. 70, a programu RUSZT na str. 78.

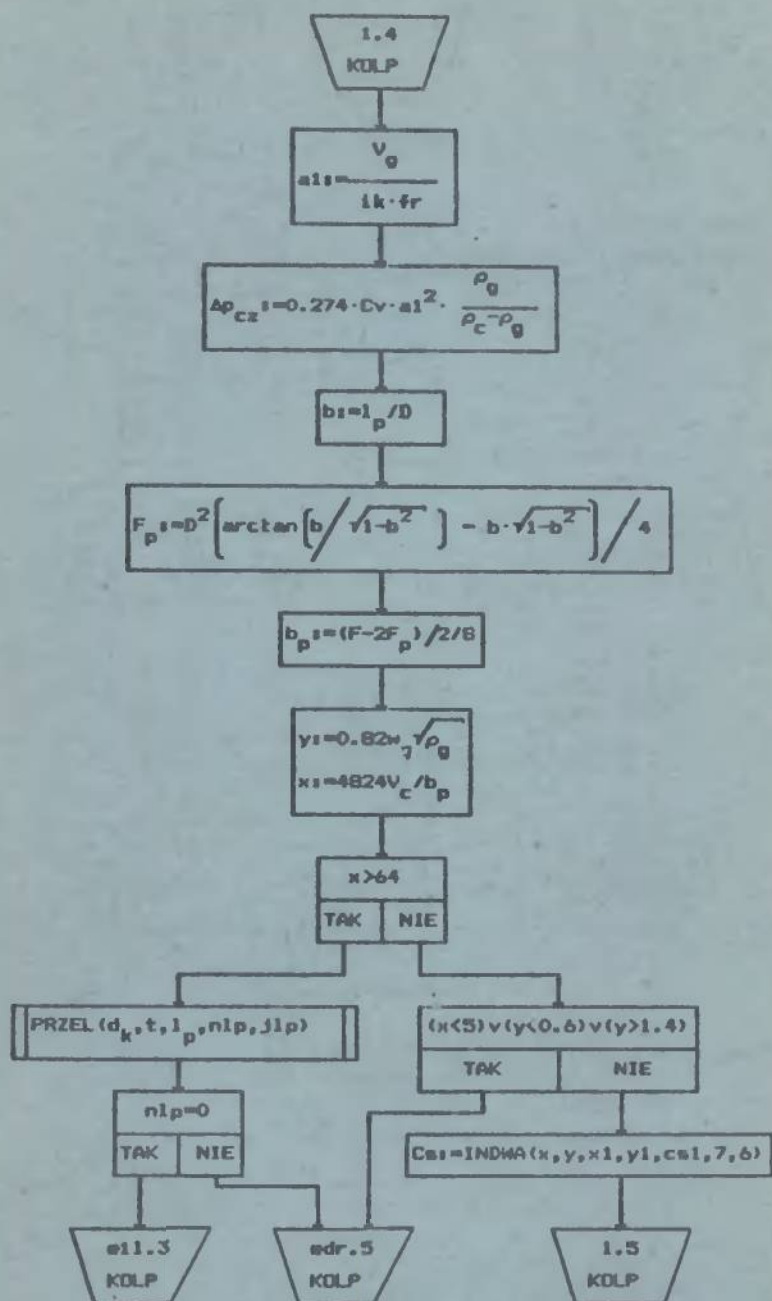
5. SCHEMATY BLOKOWE PROGRAMÓW

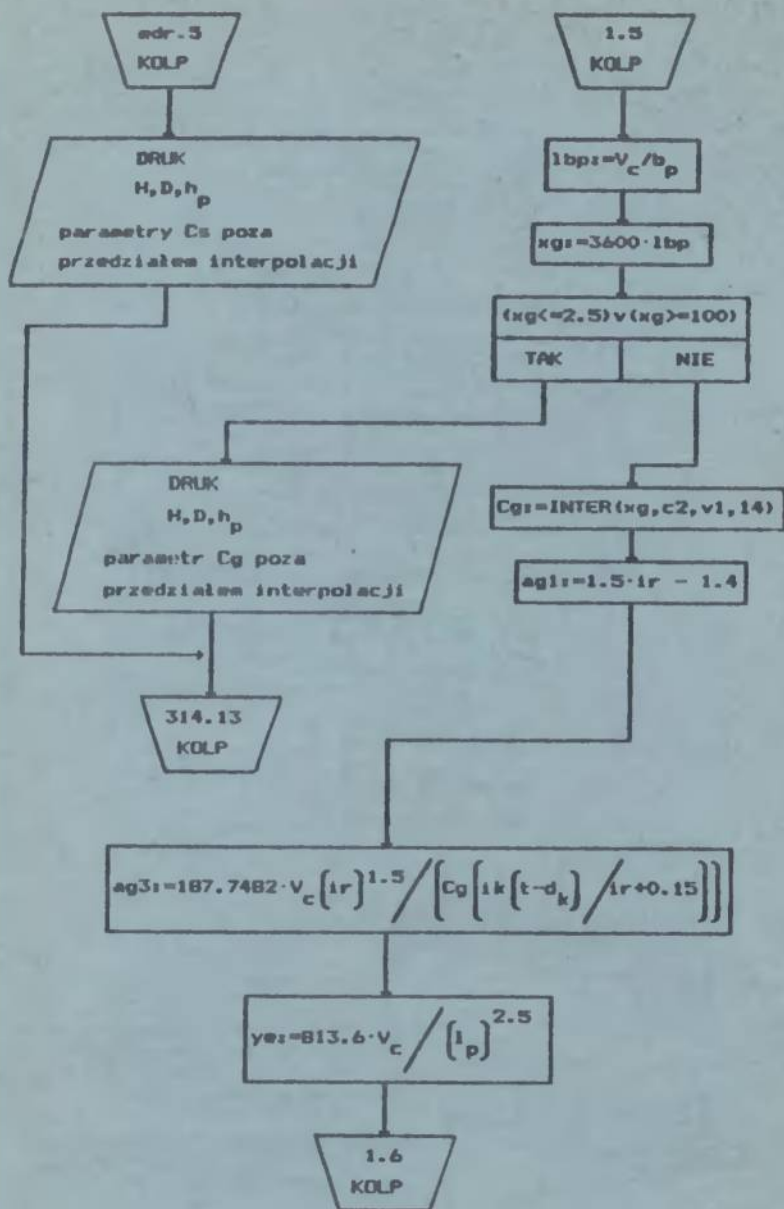
3.1. Schemat blokowy programu KOLP

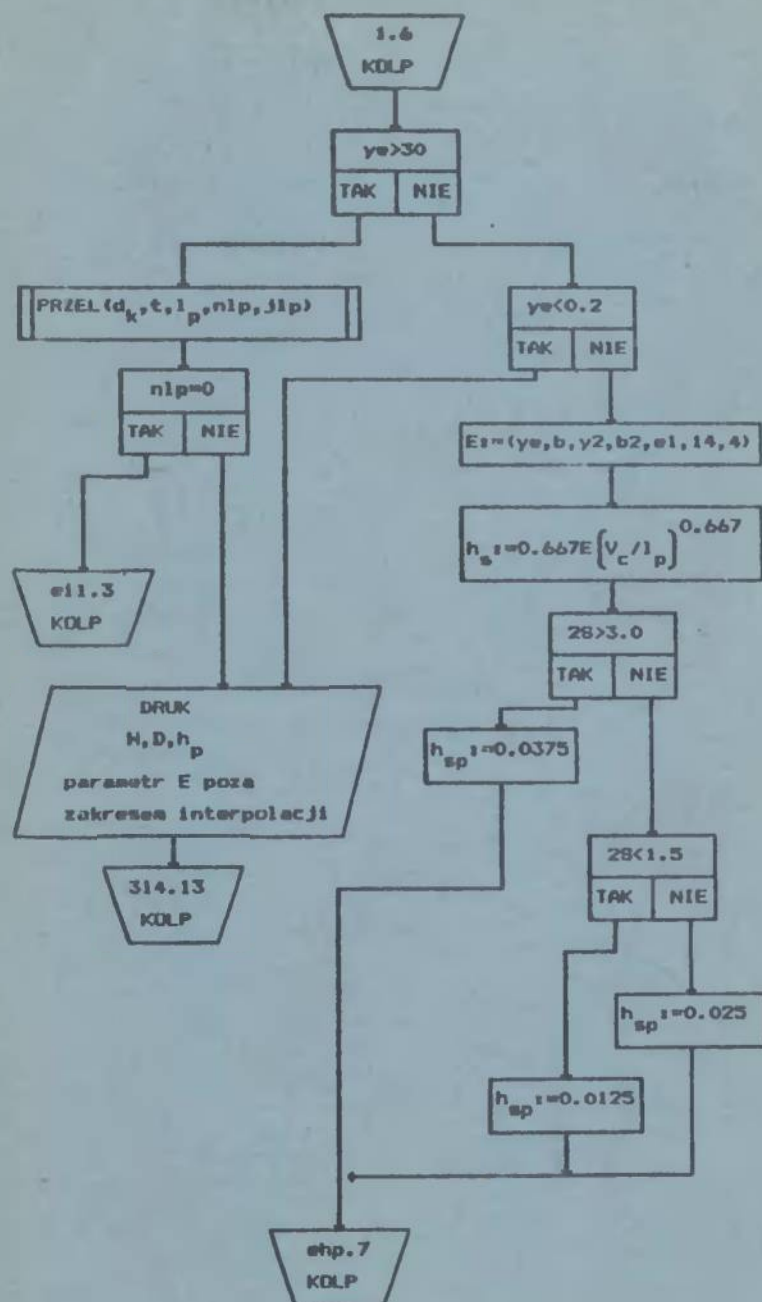


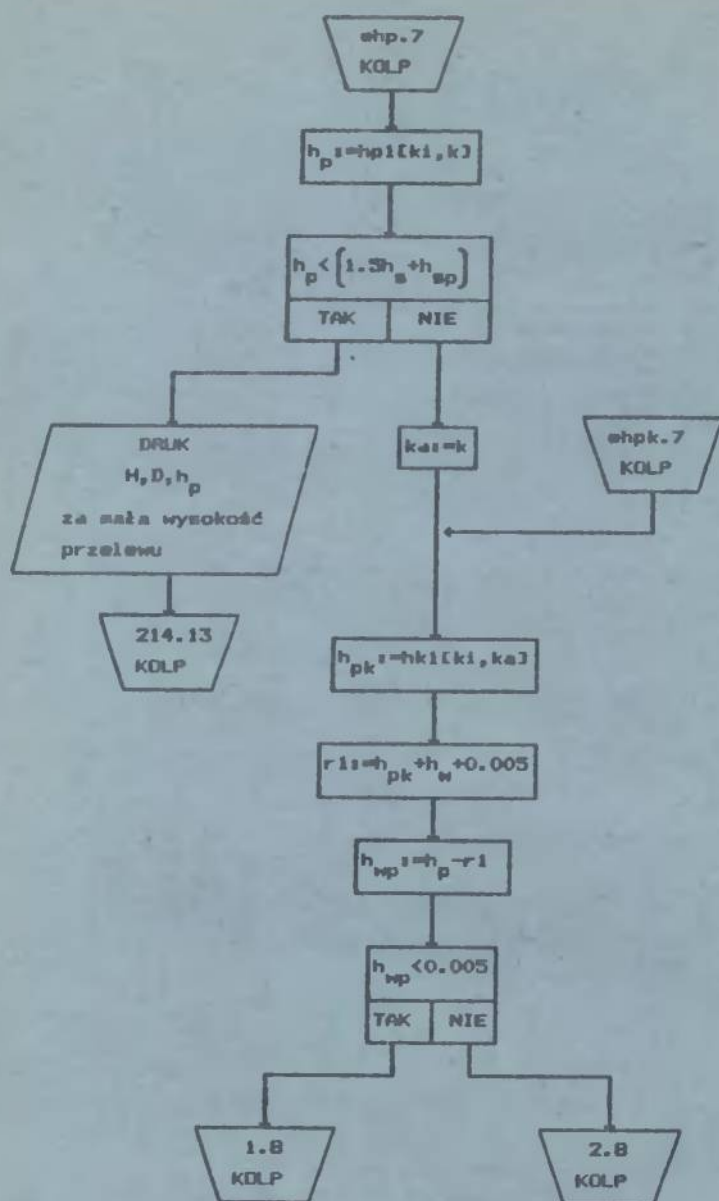


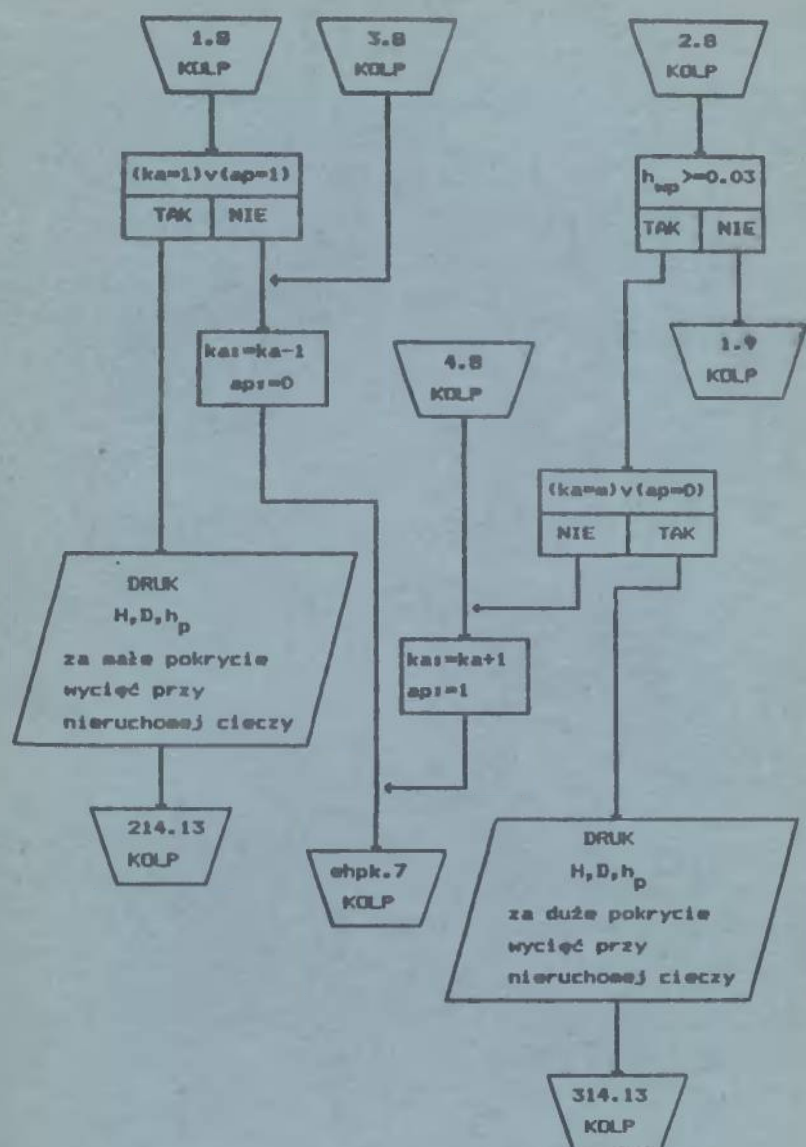


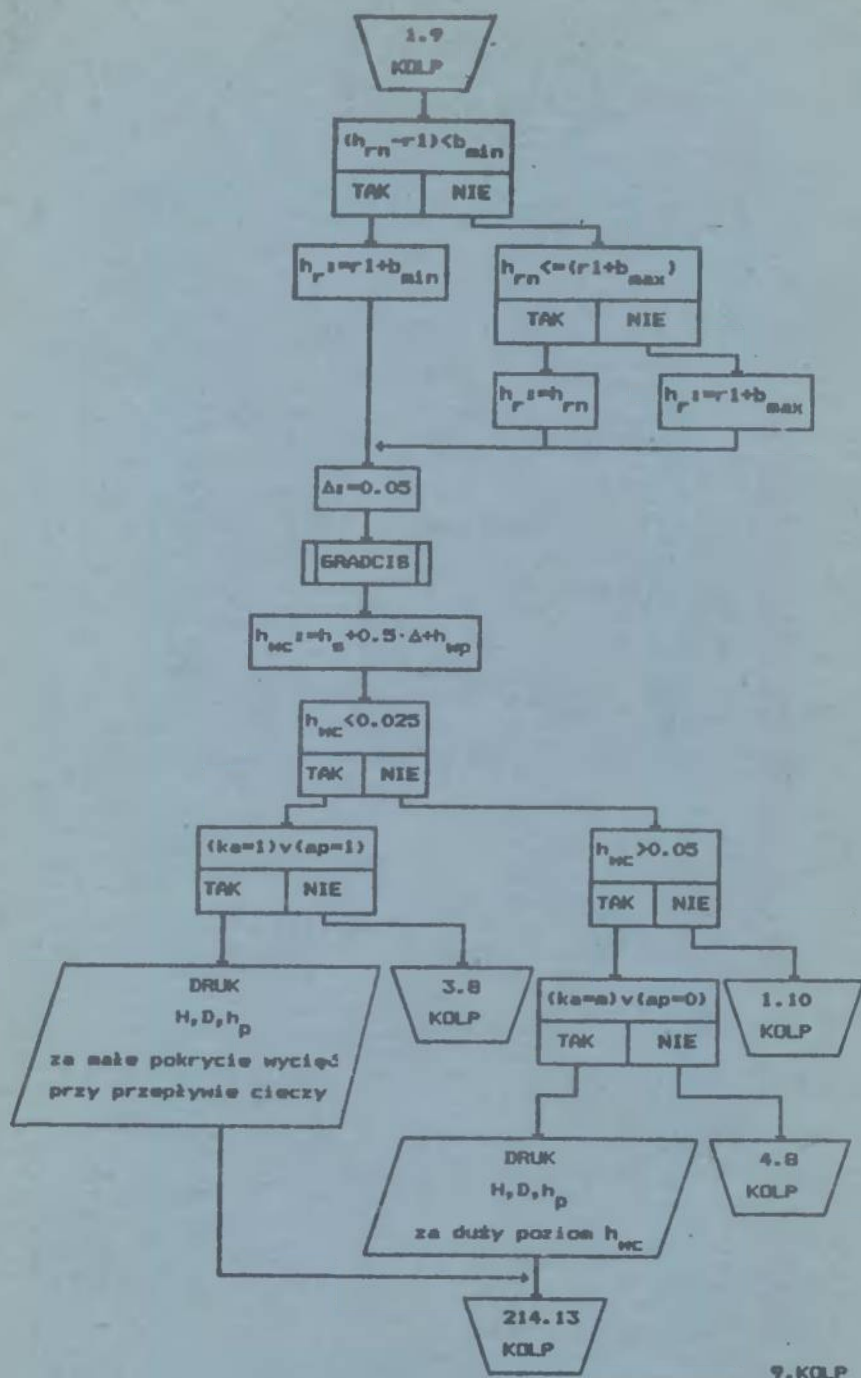


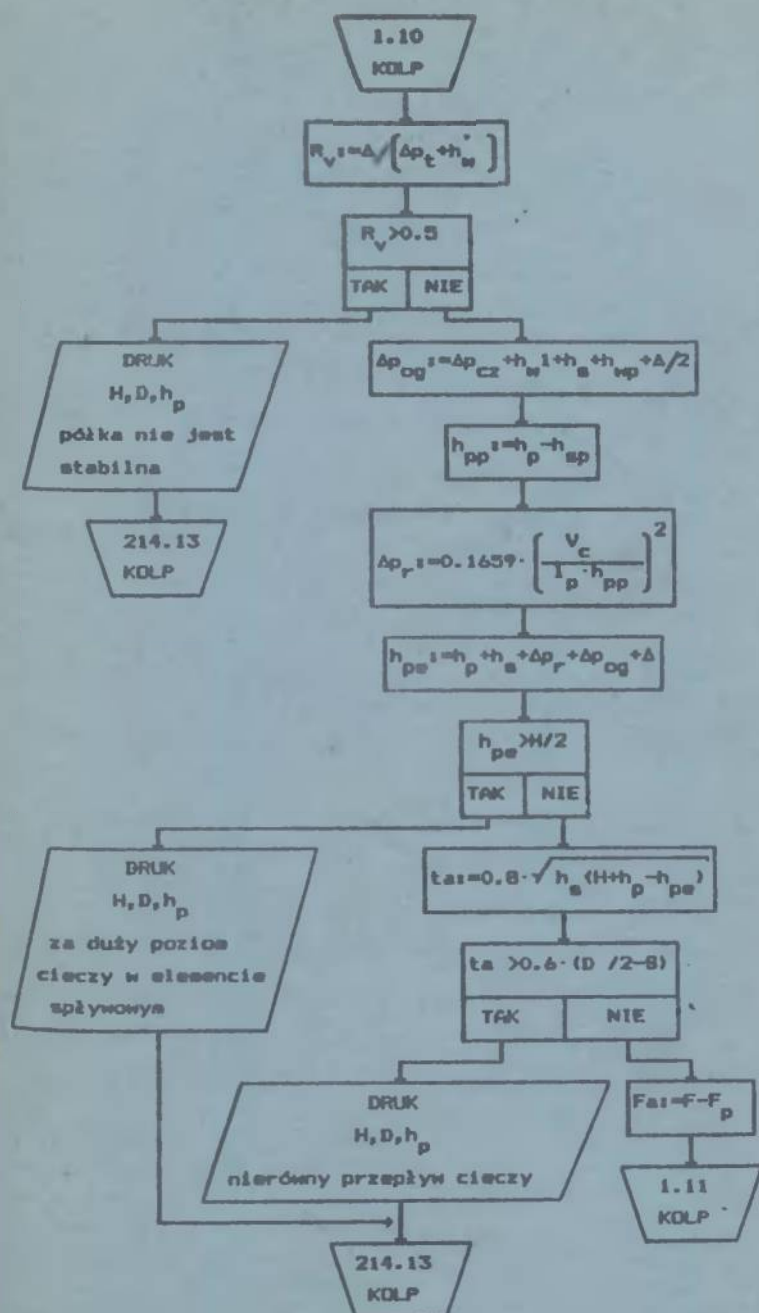


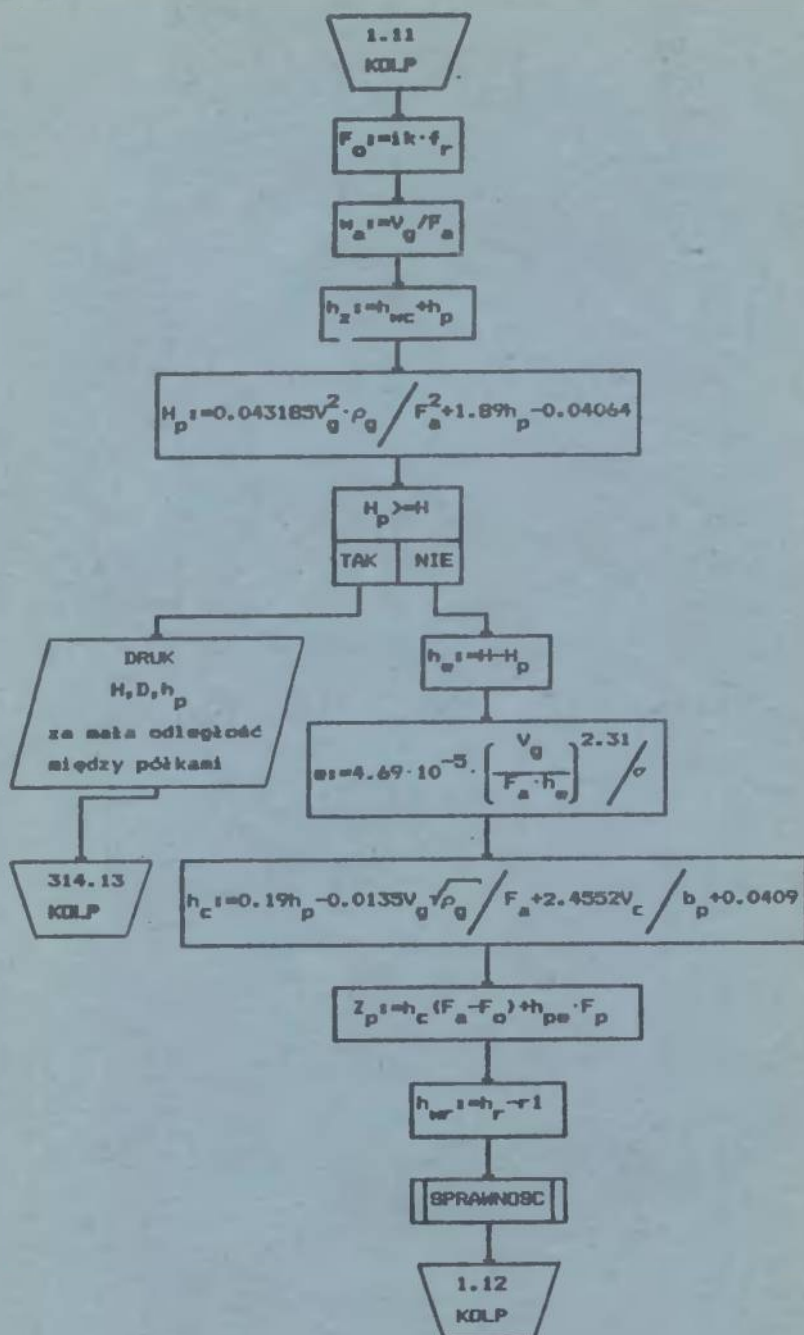


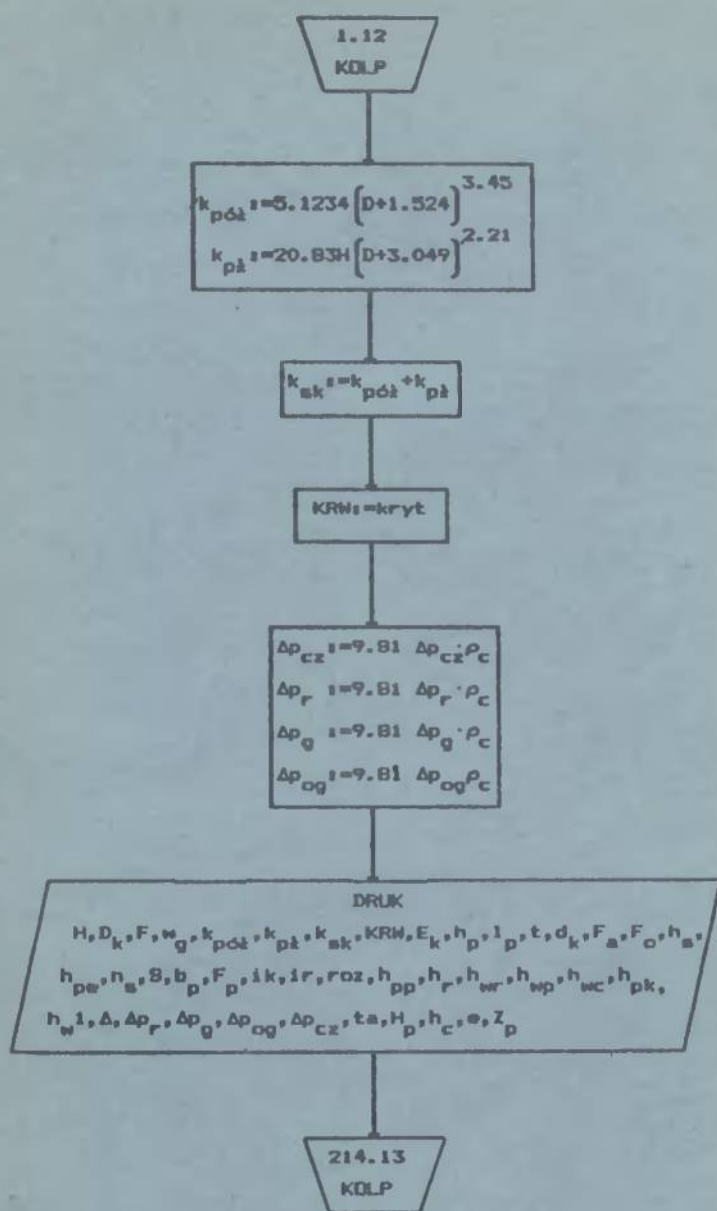


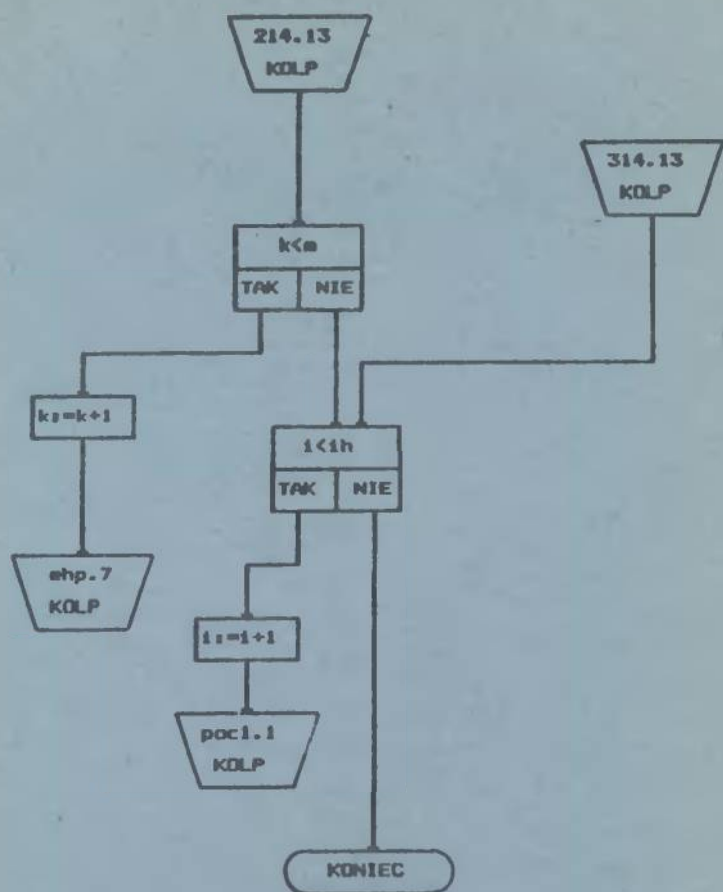




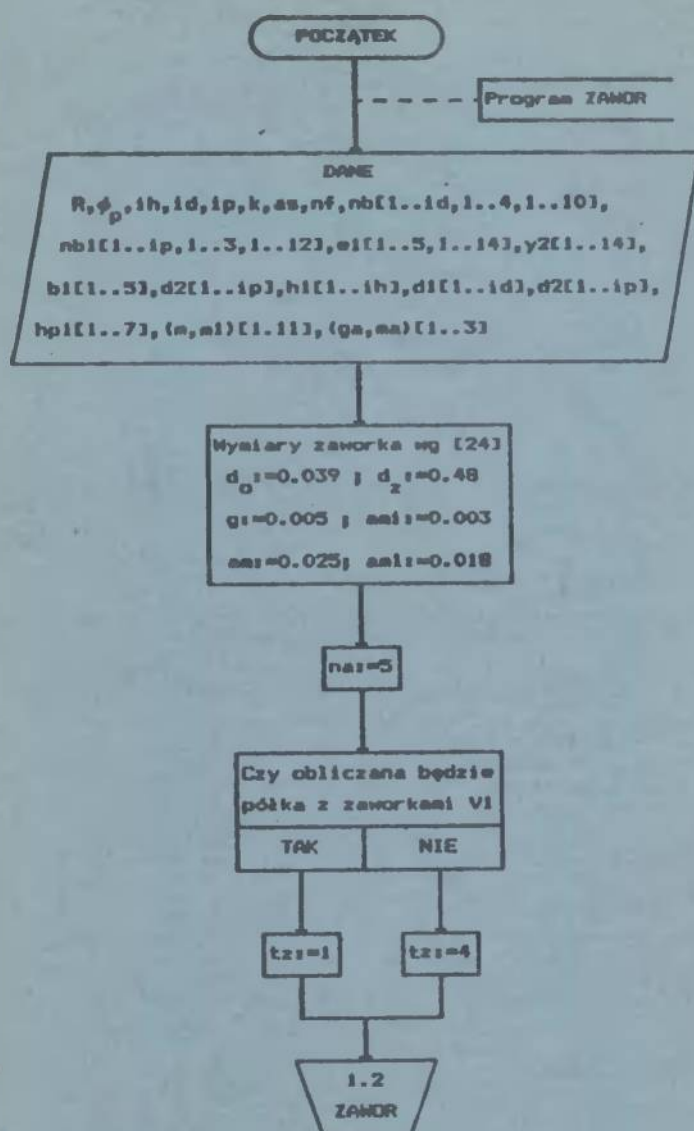


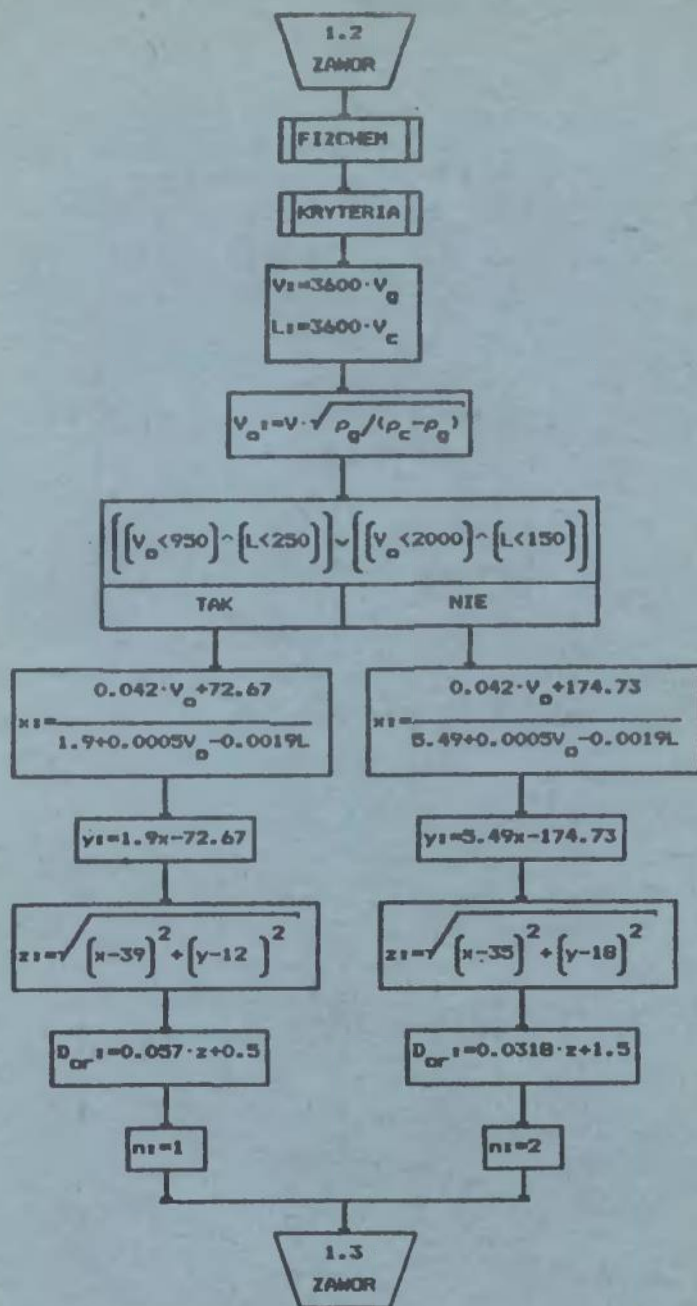


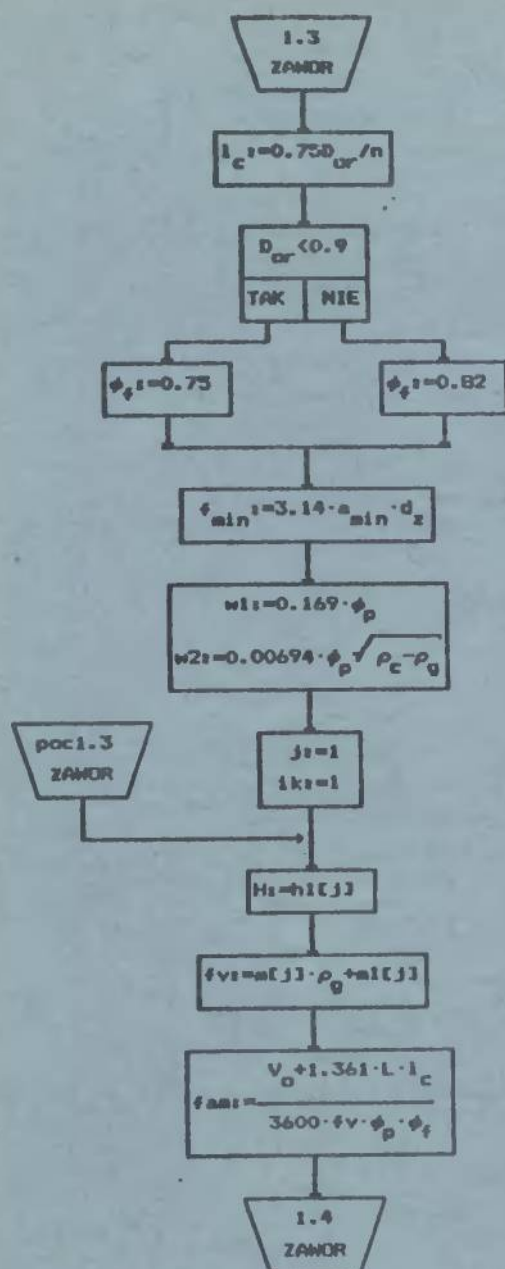


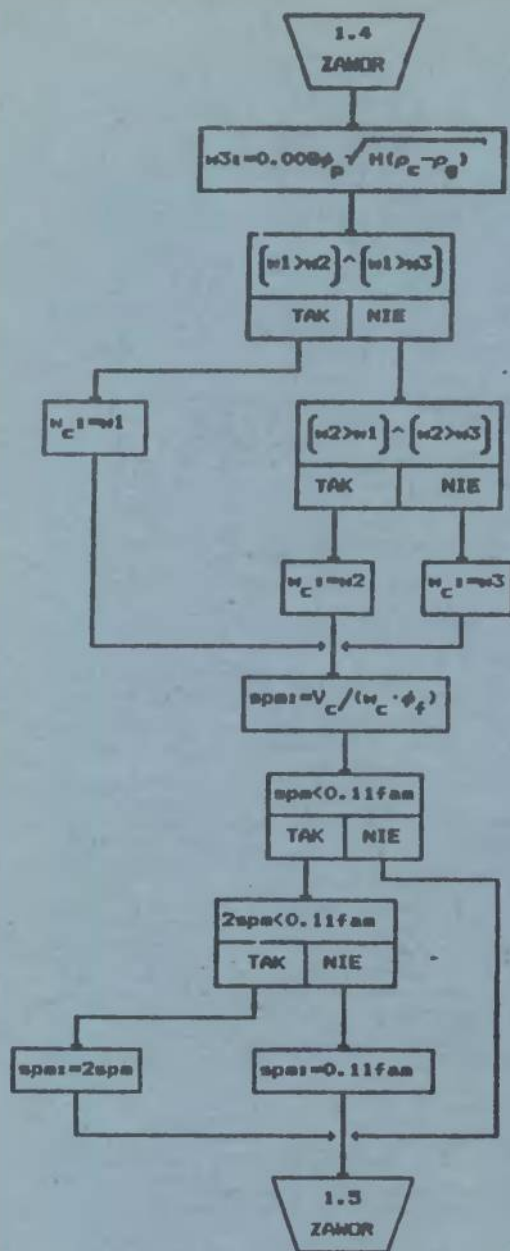


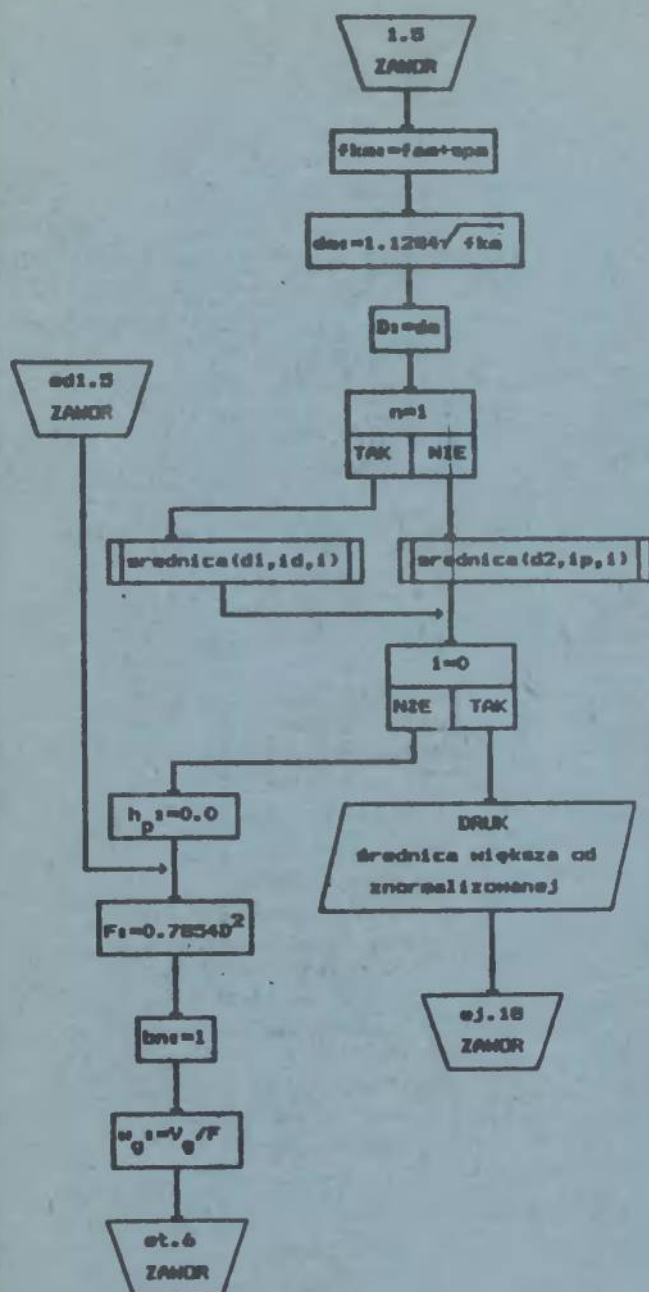
3.2. Schemat blokowy programu ZAMOR

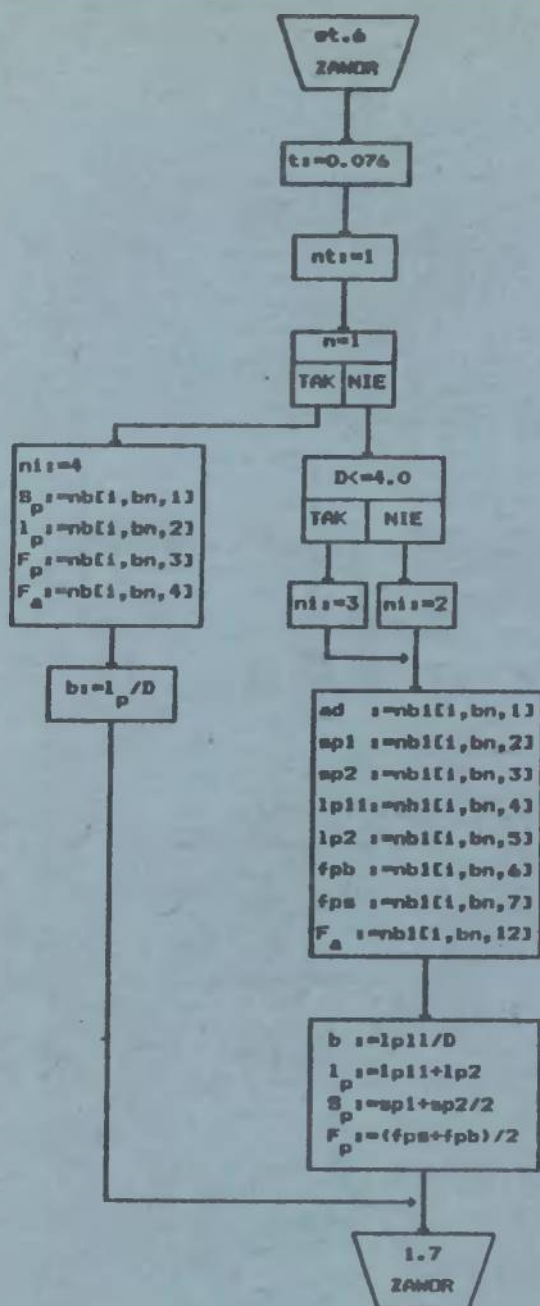


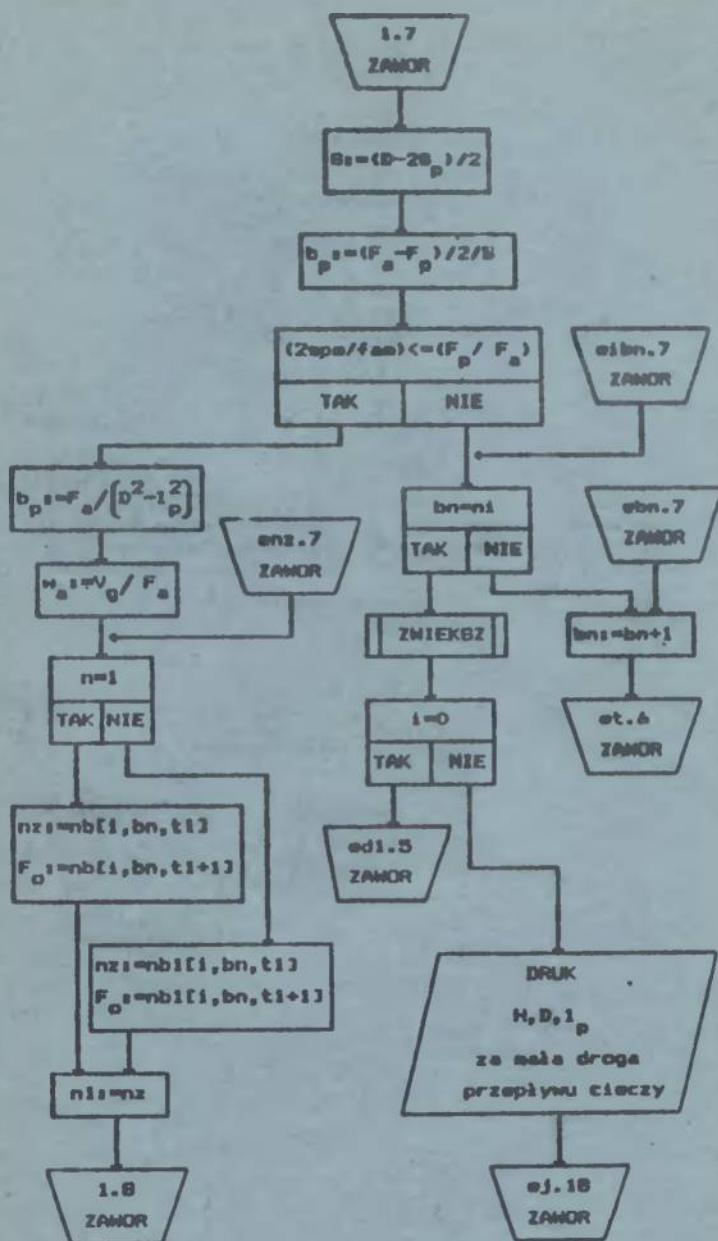


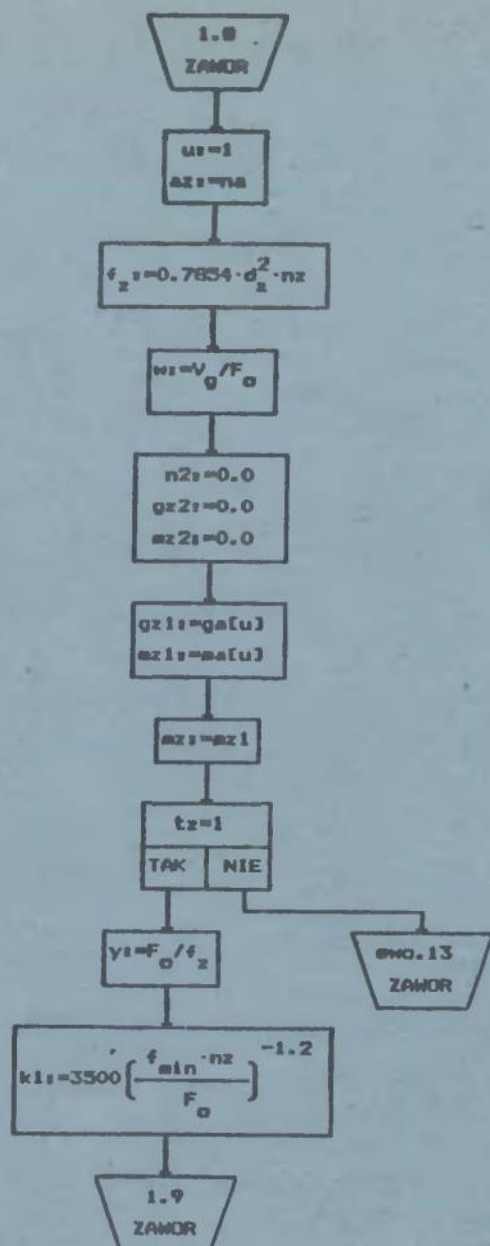


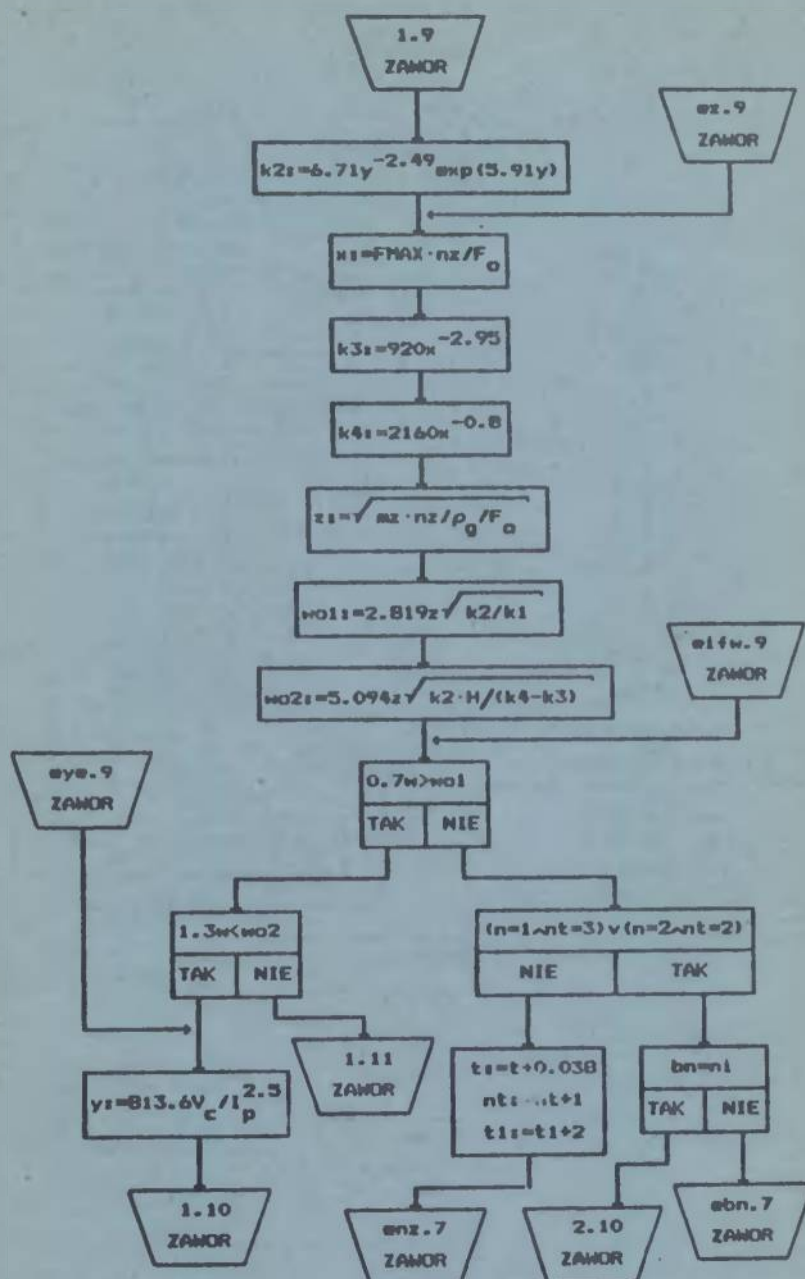


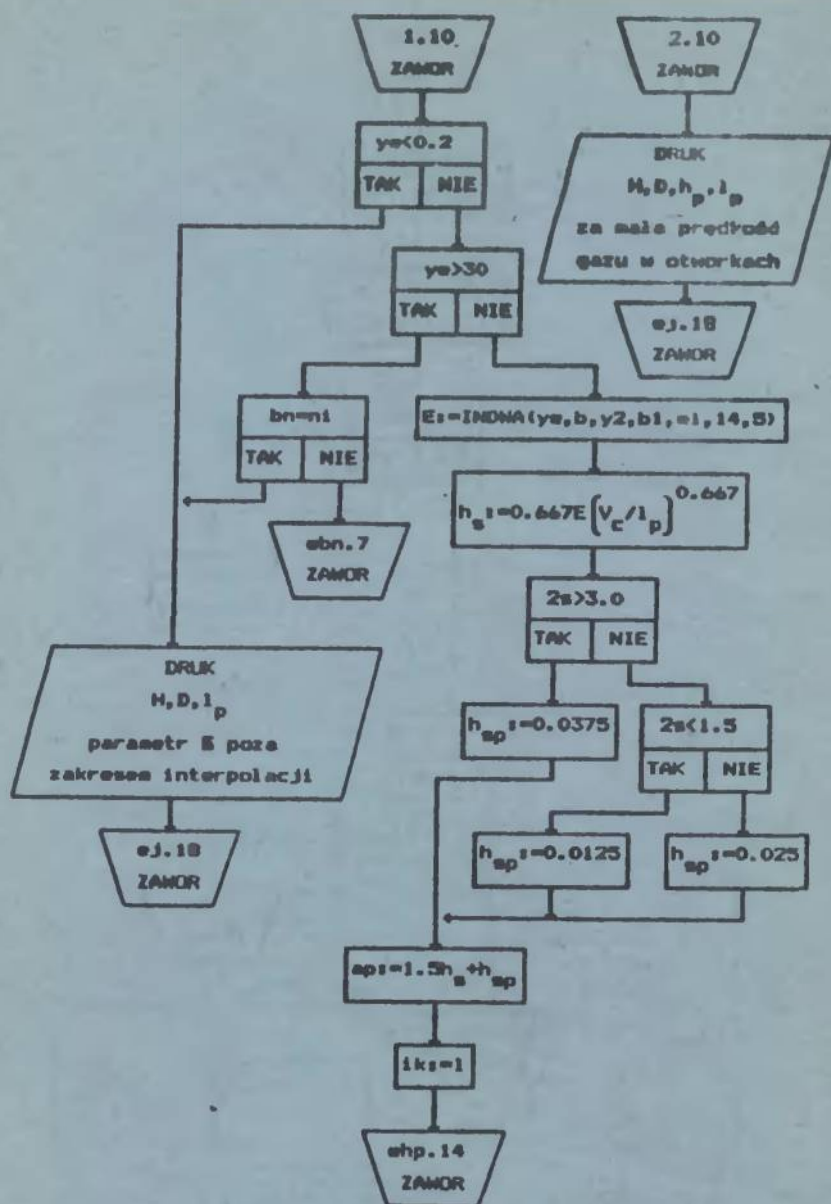


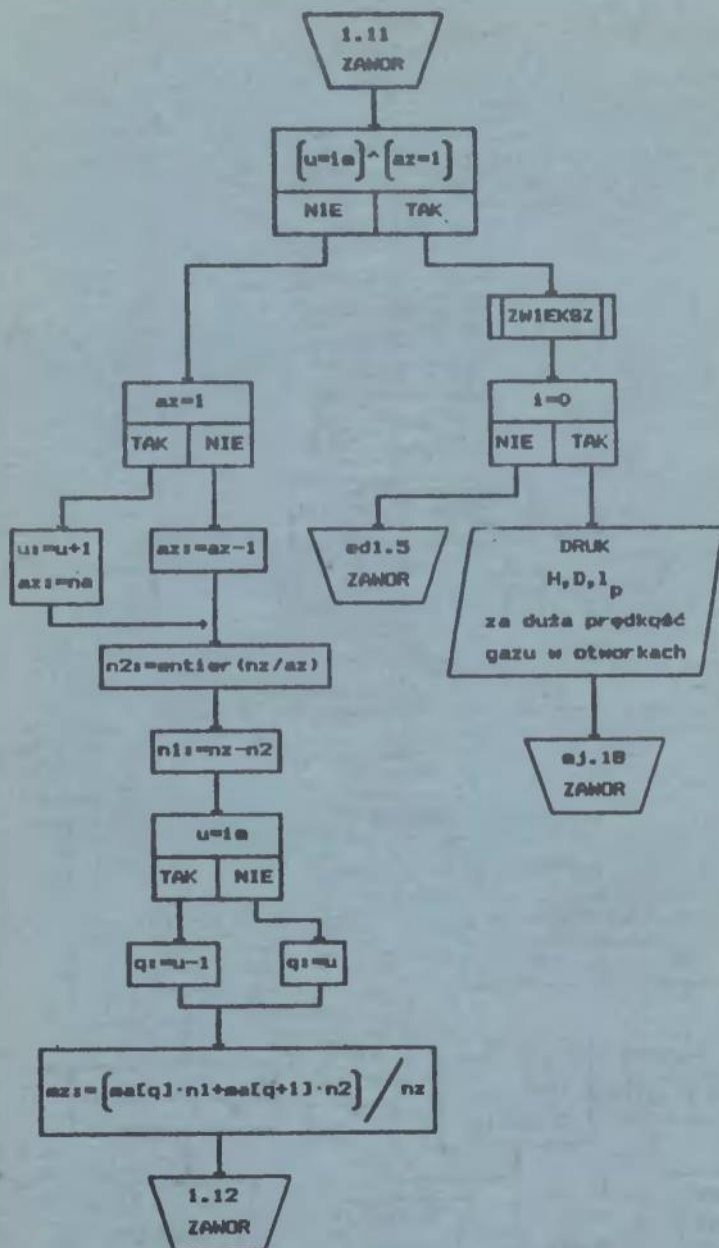


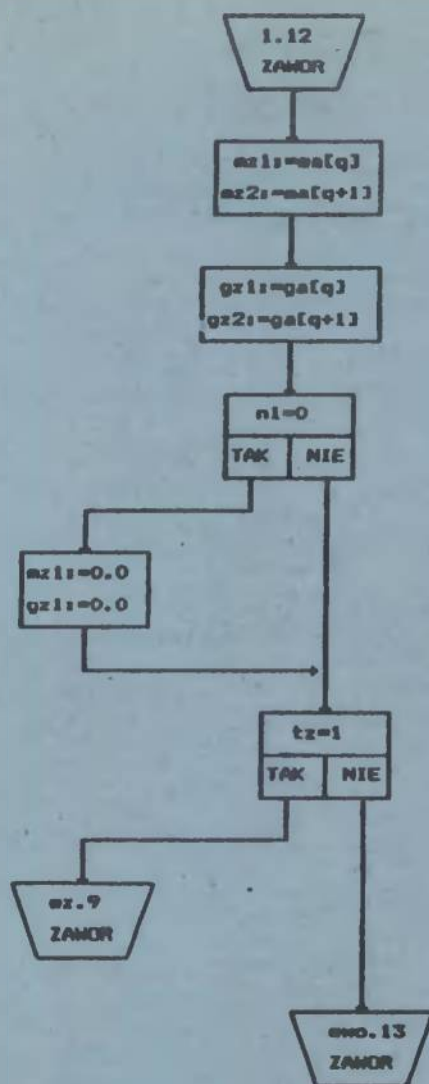


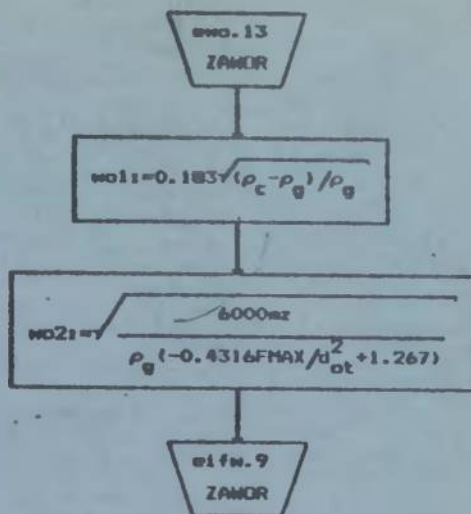


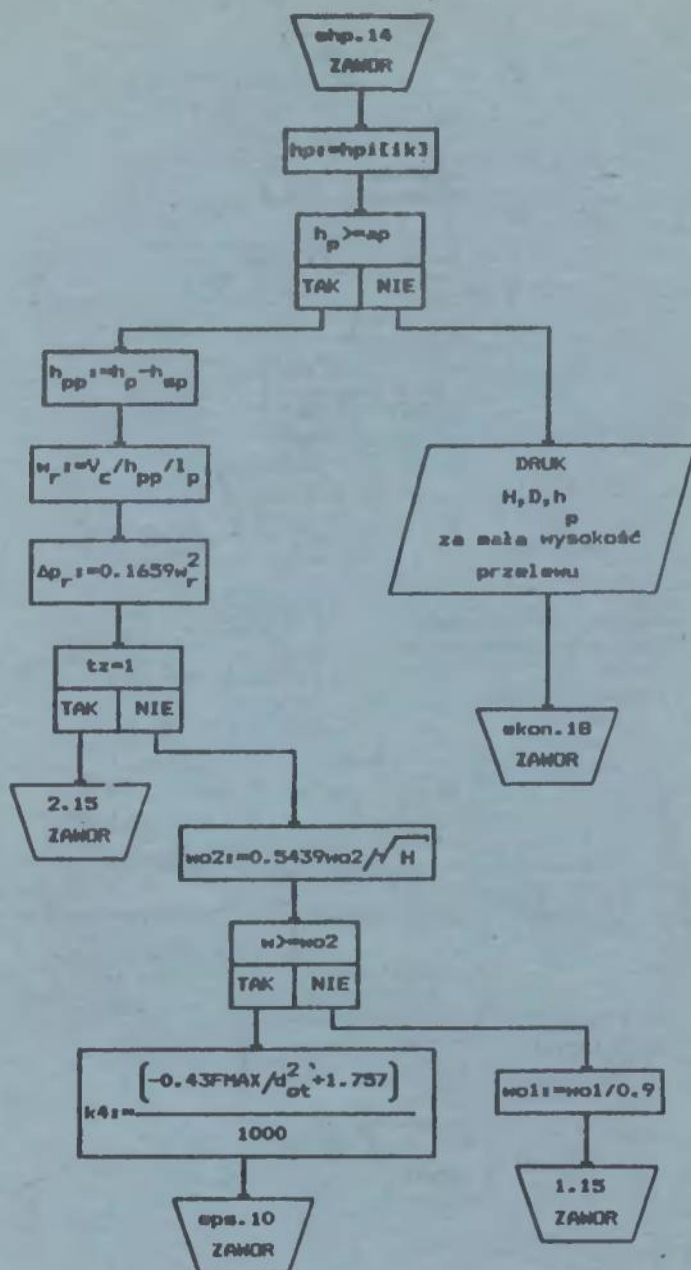


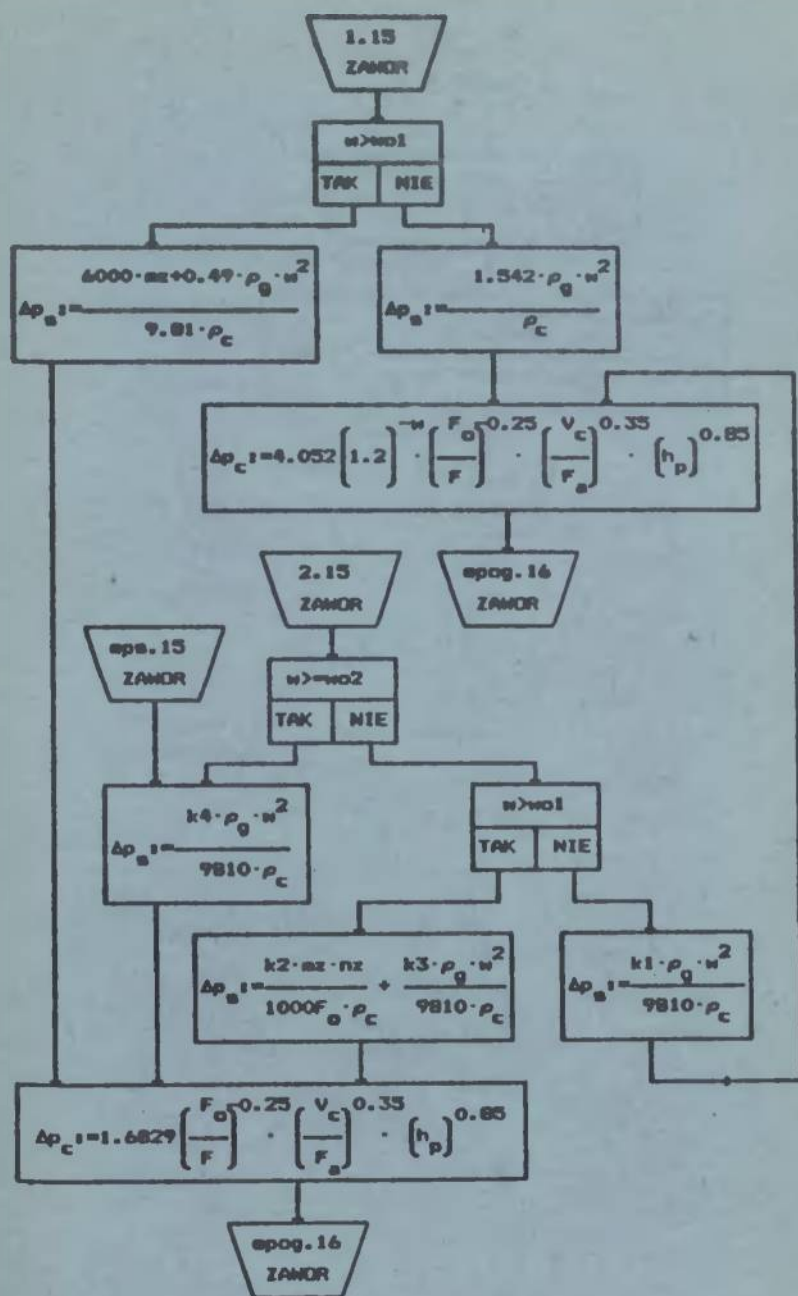


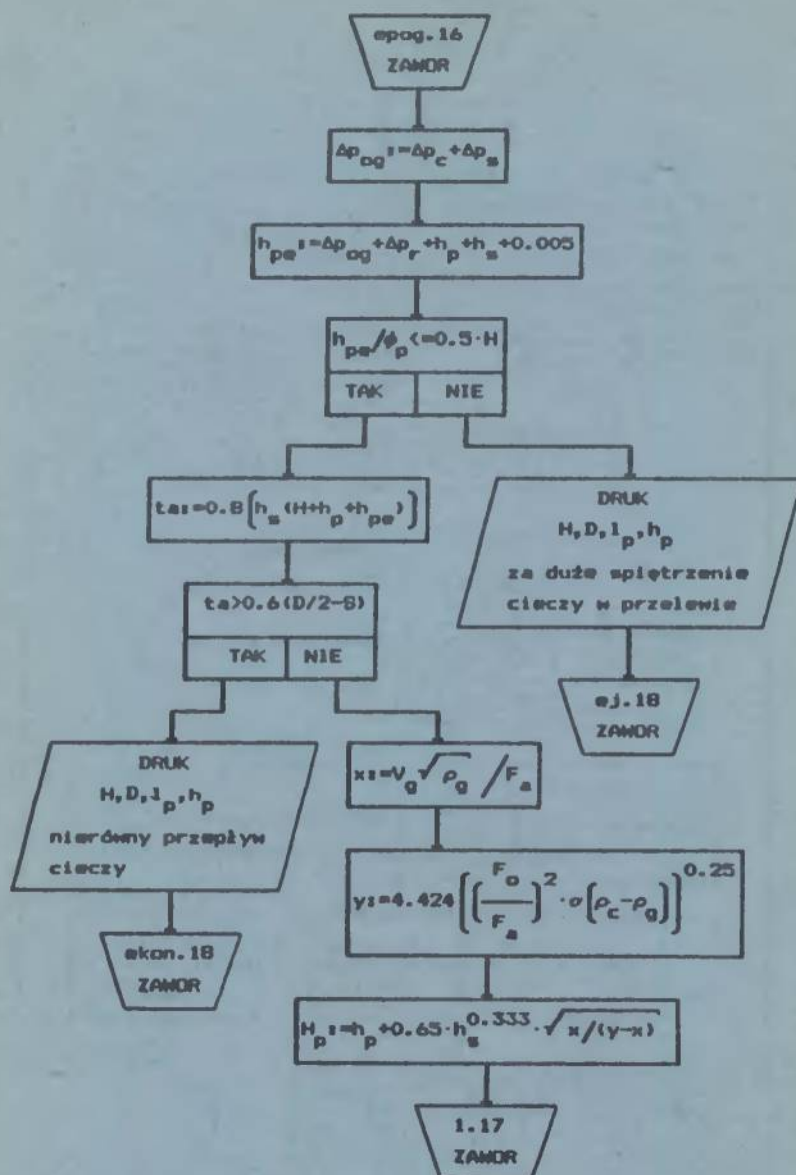


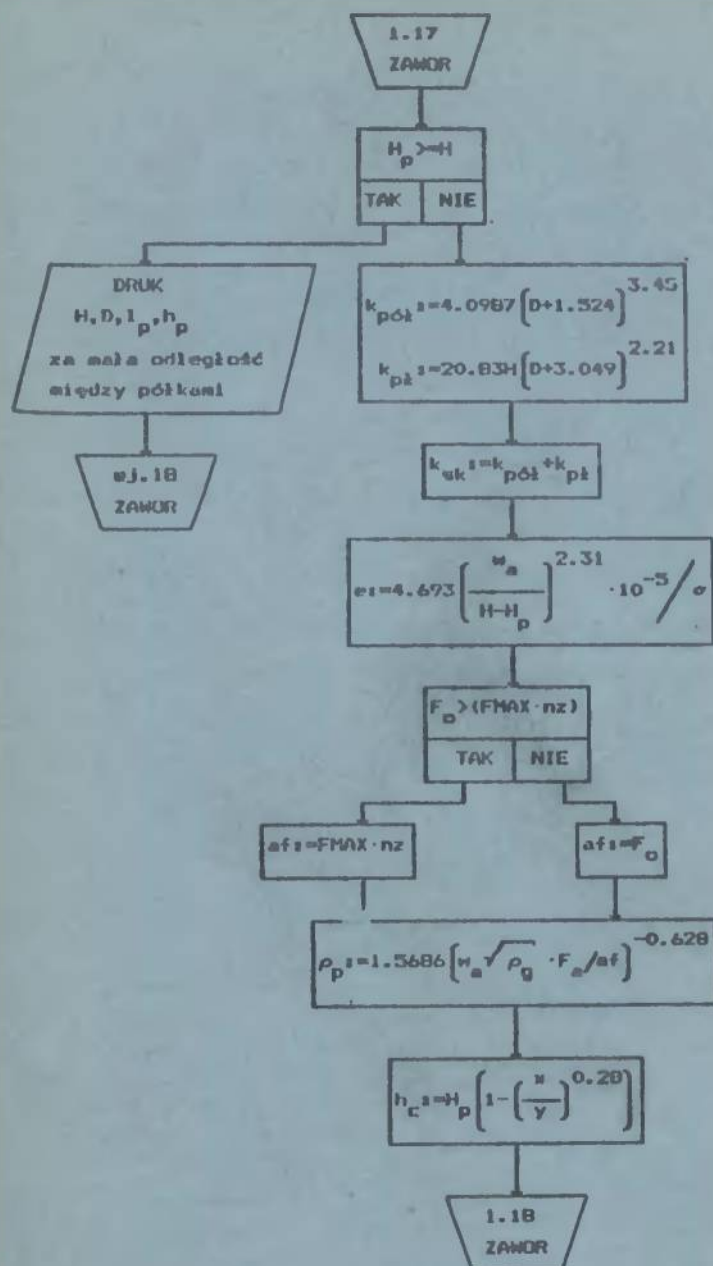


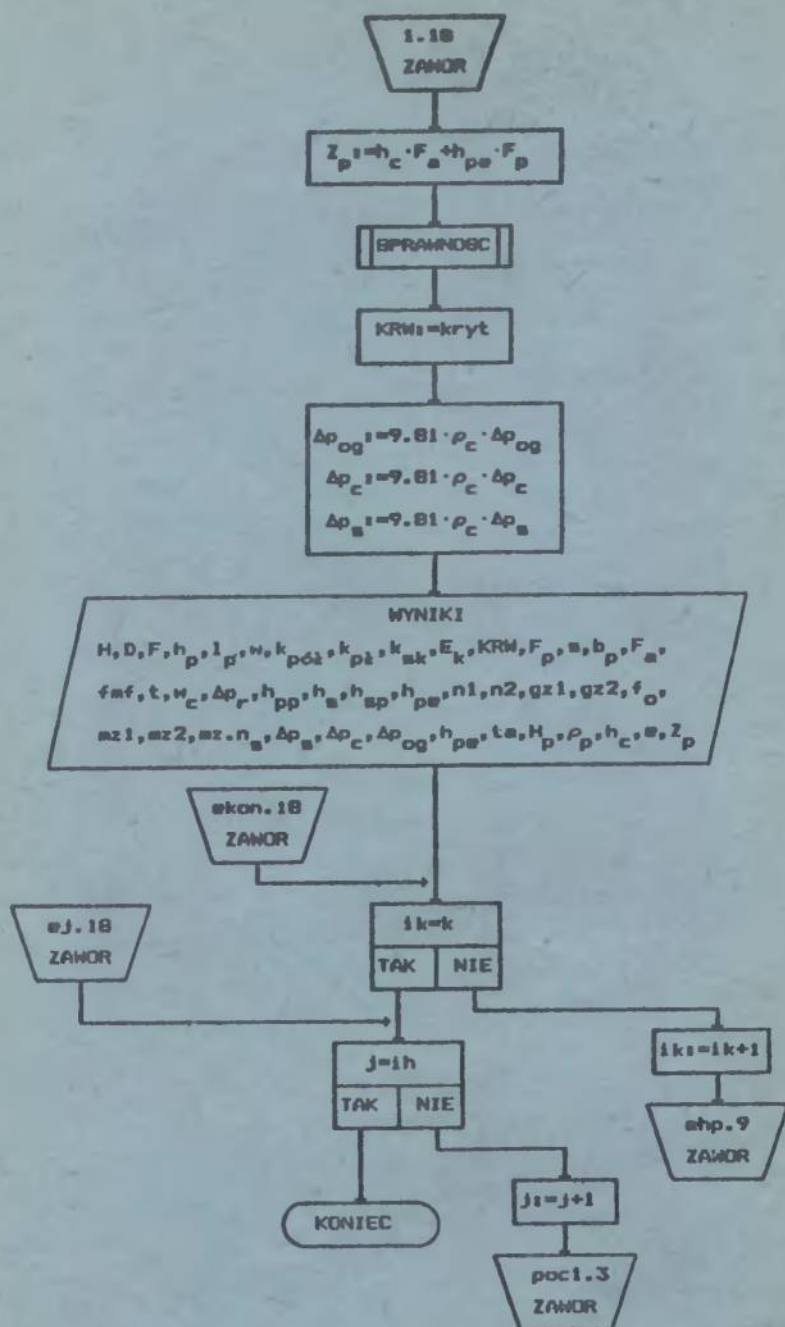




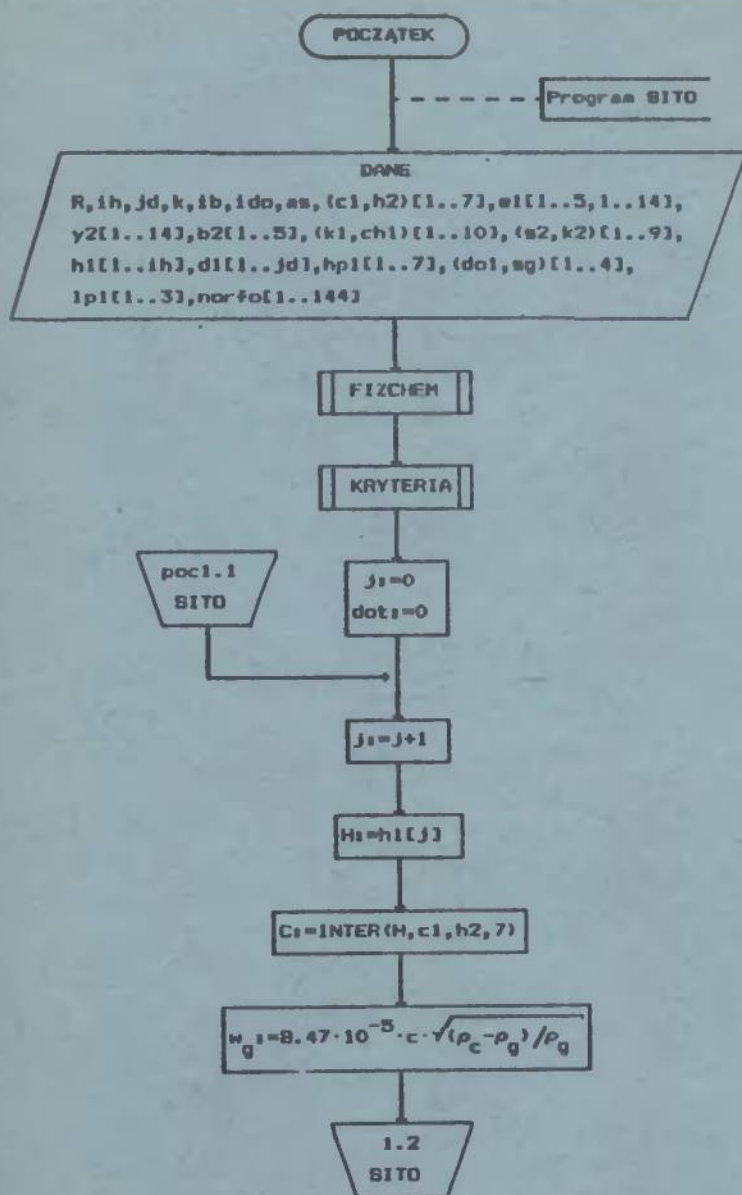


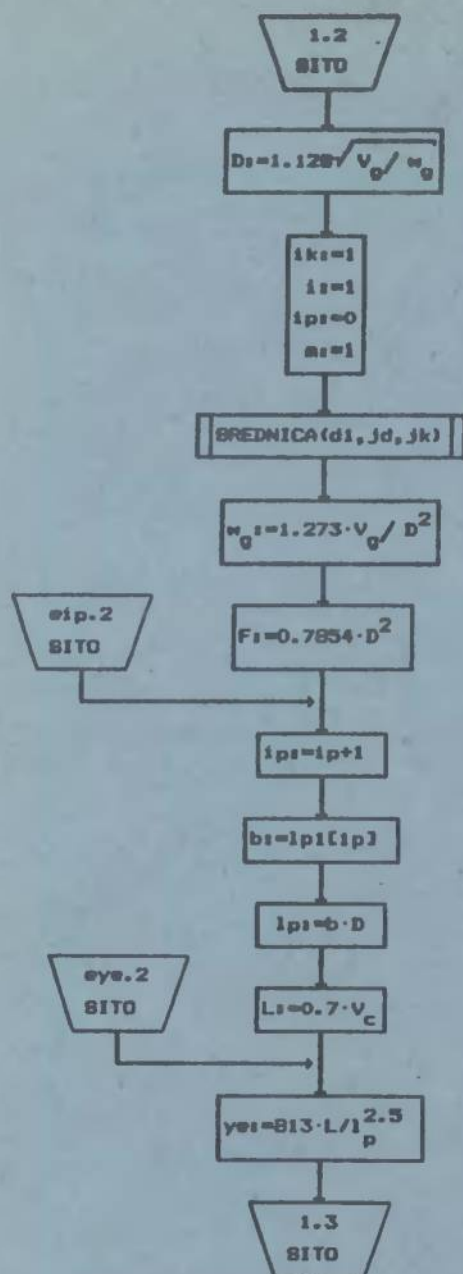


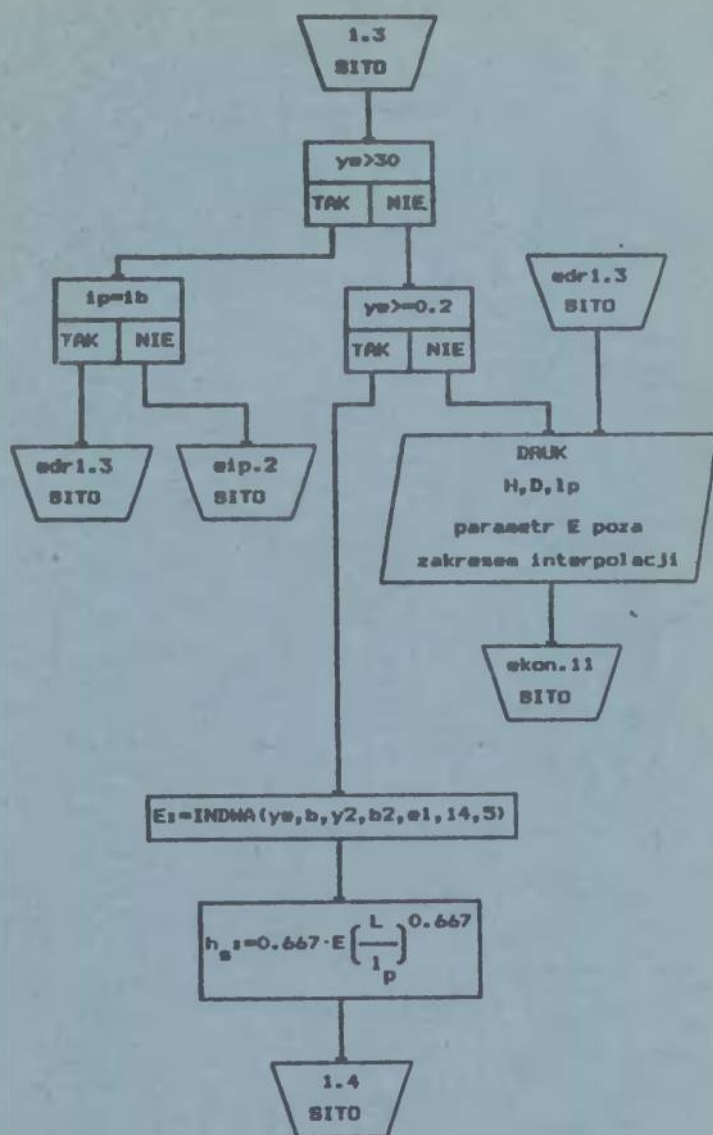


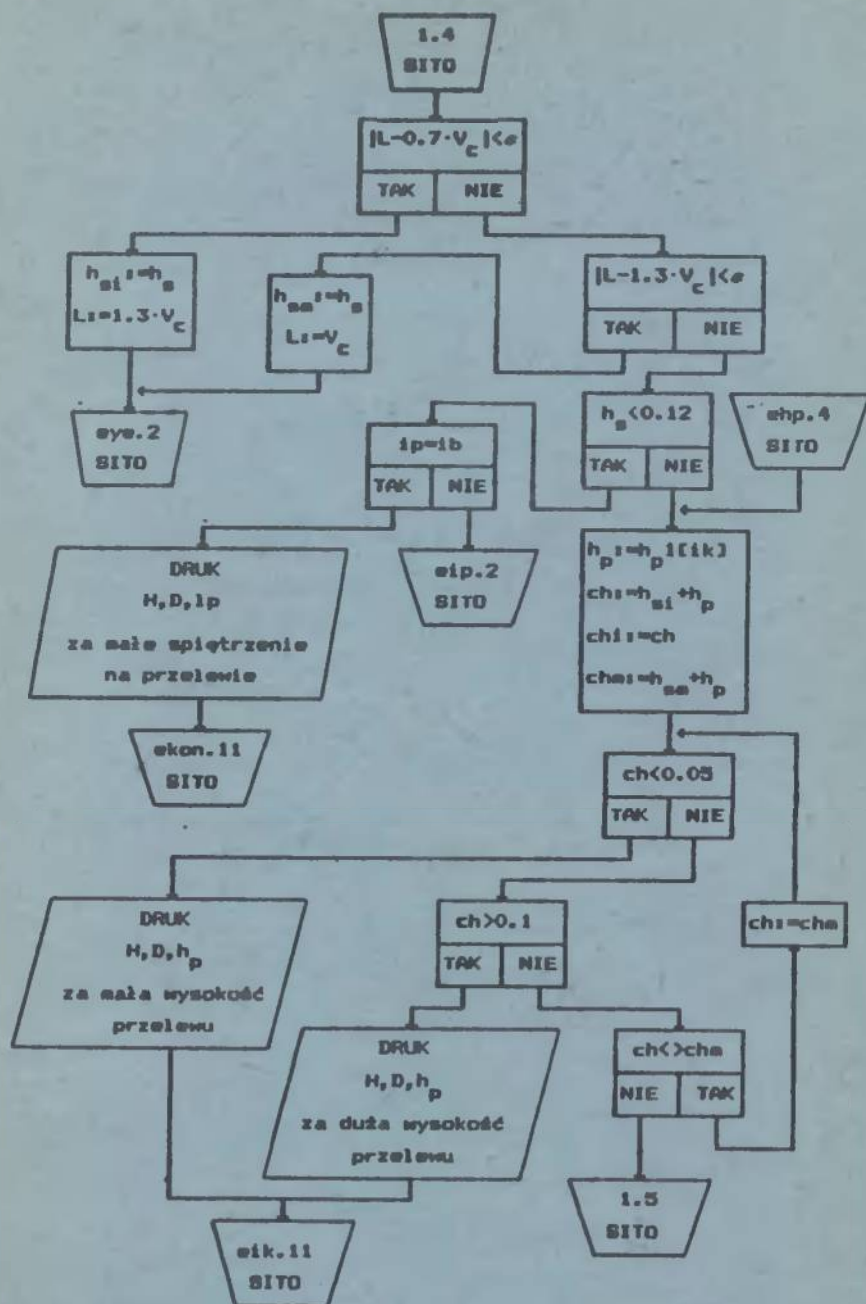


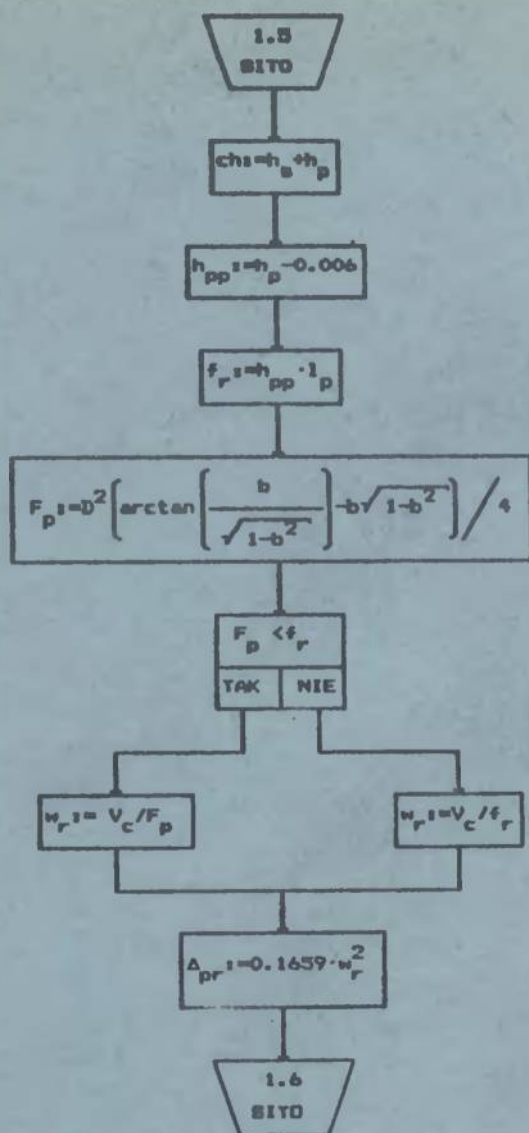
5.3. Schemat blokowy programu SITO



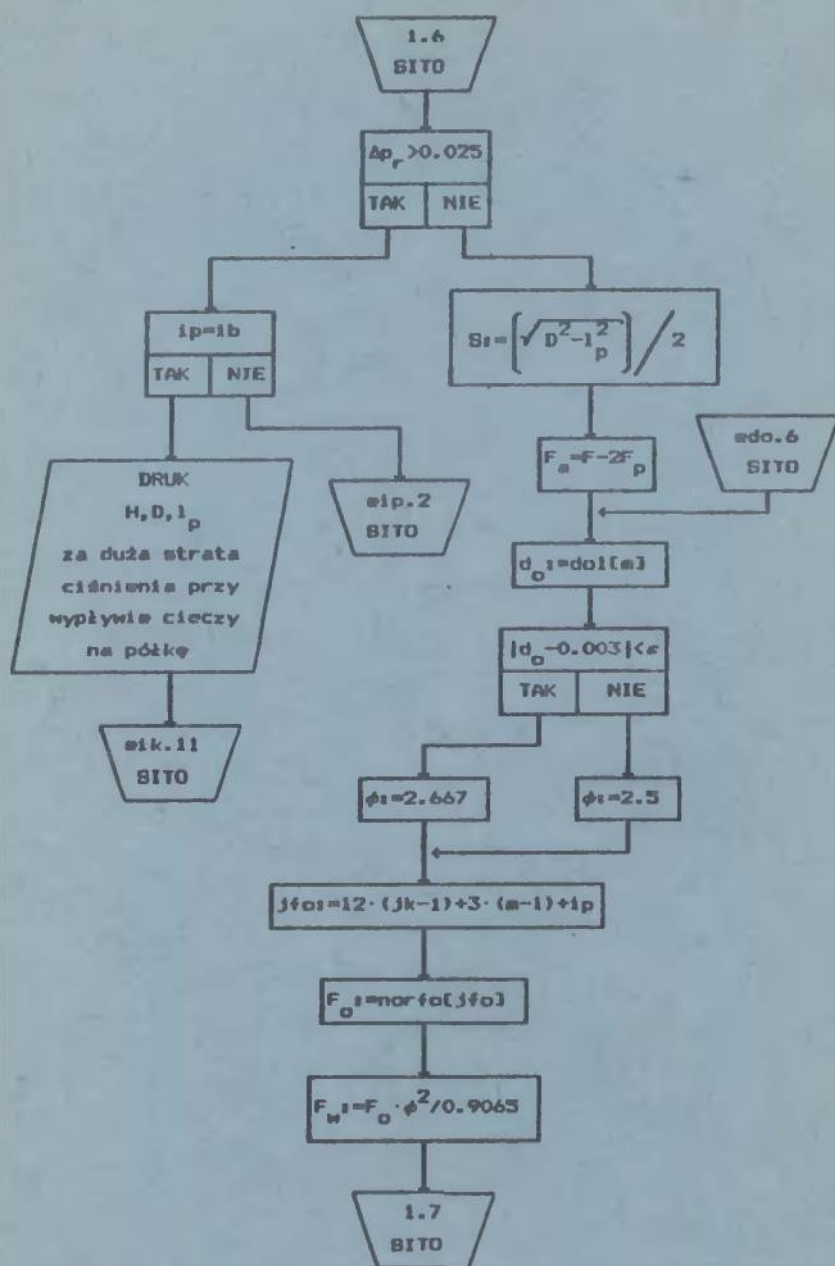


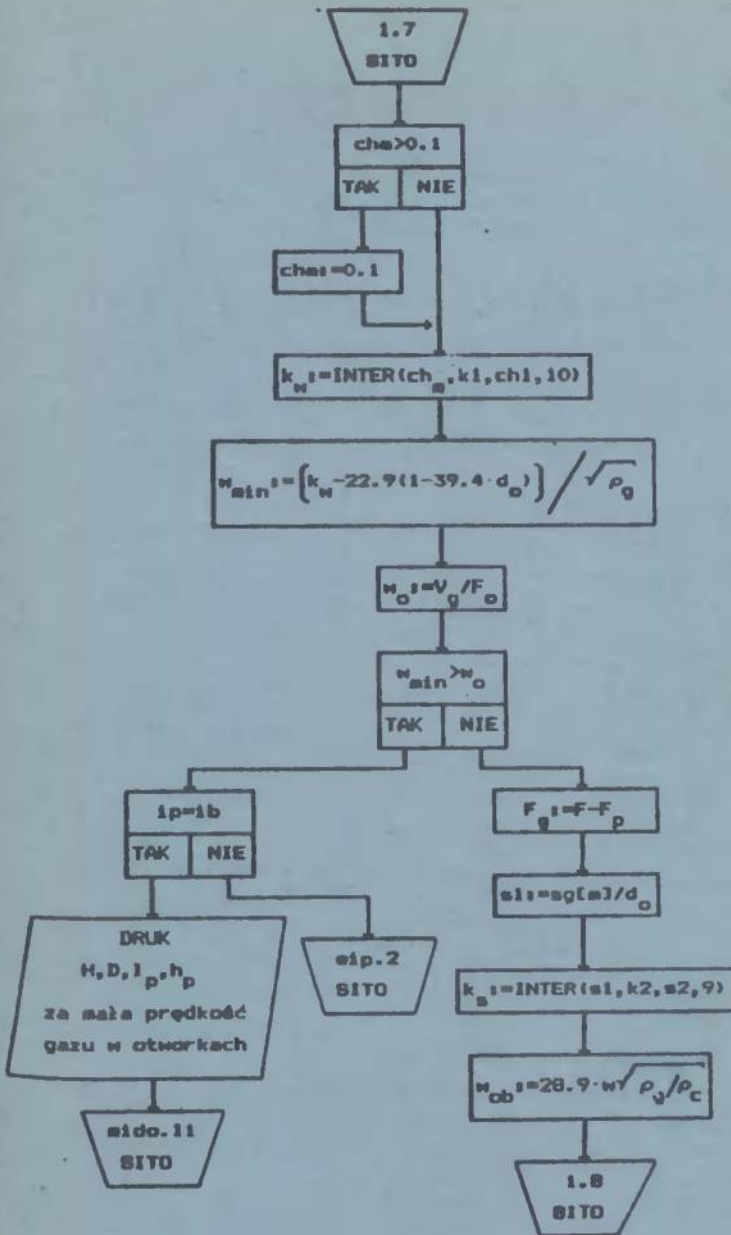


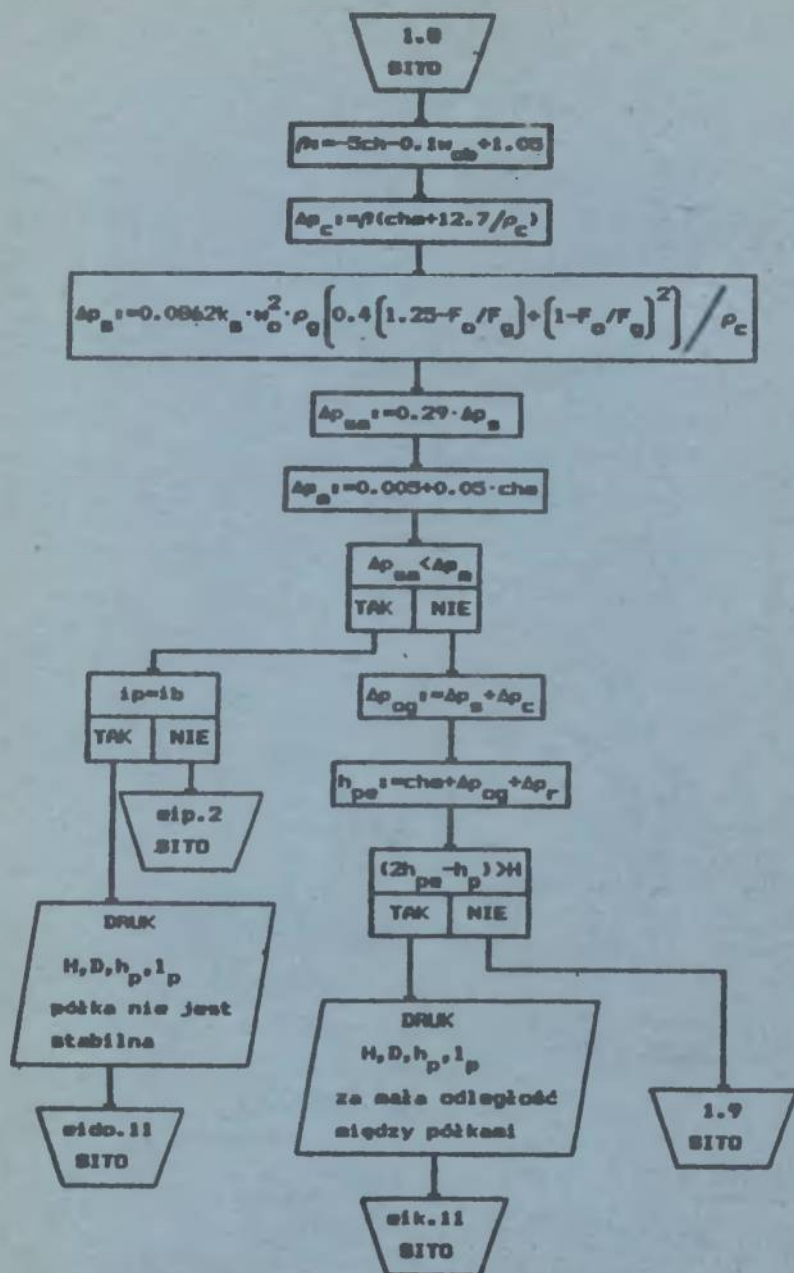


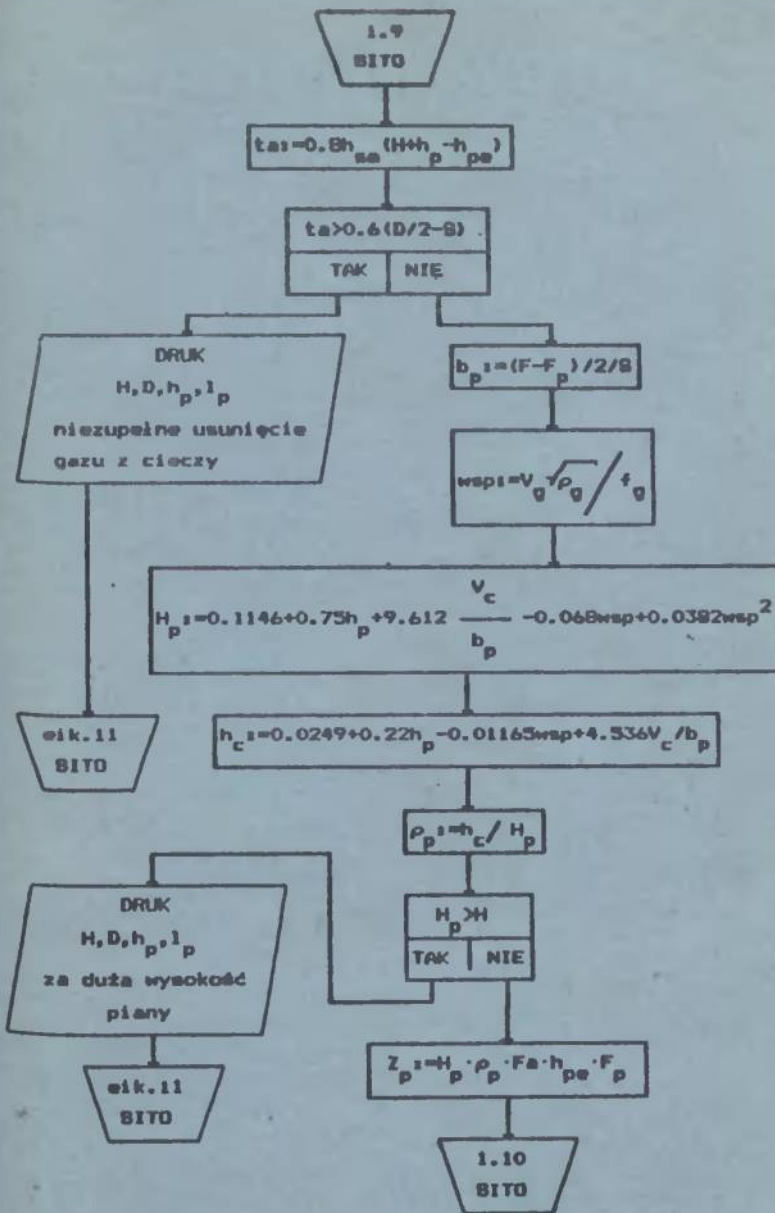


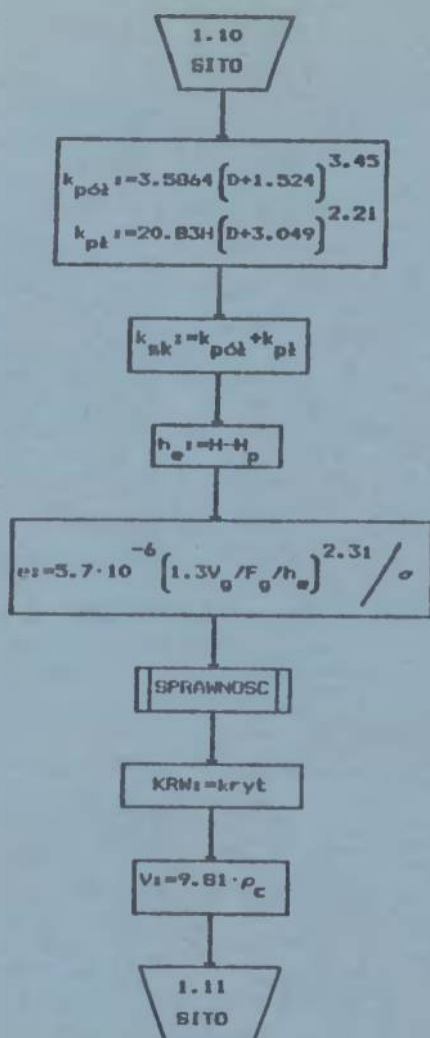
5.81 TO

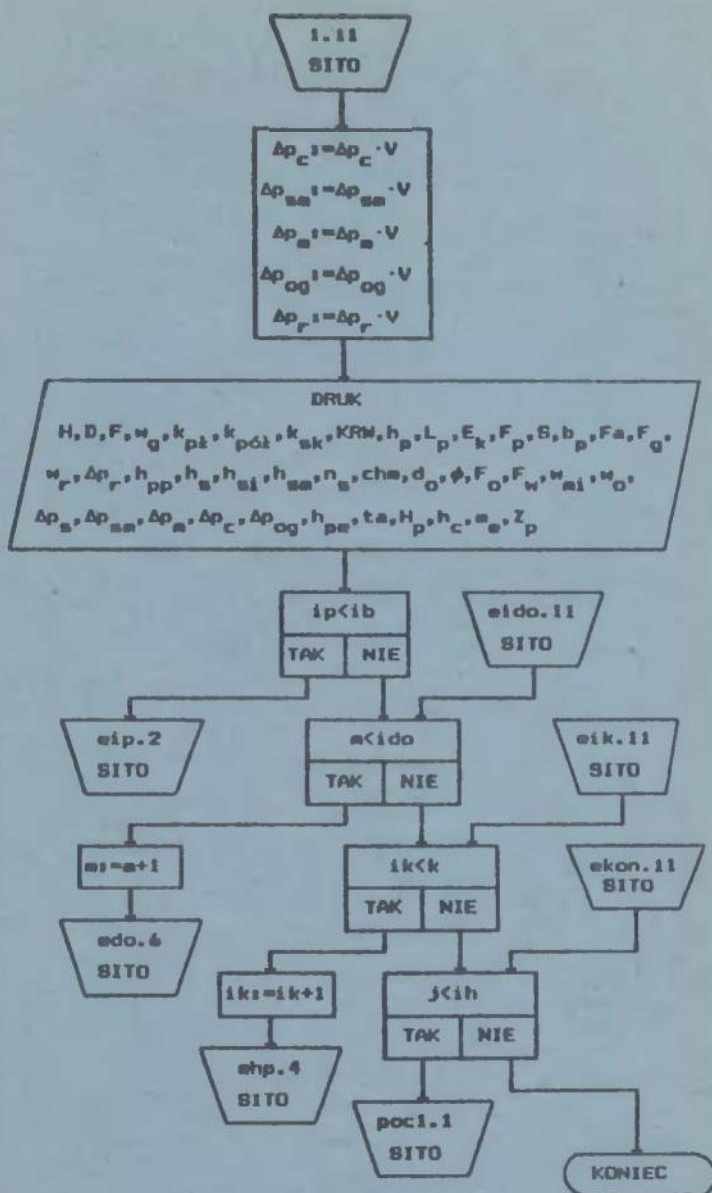




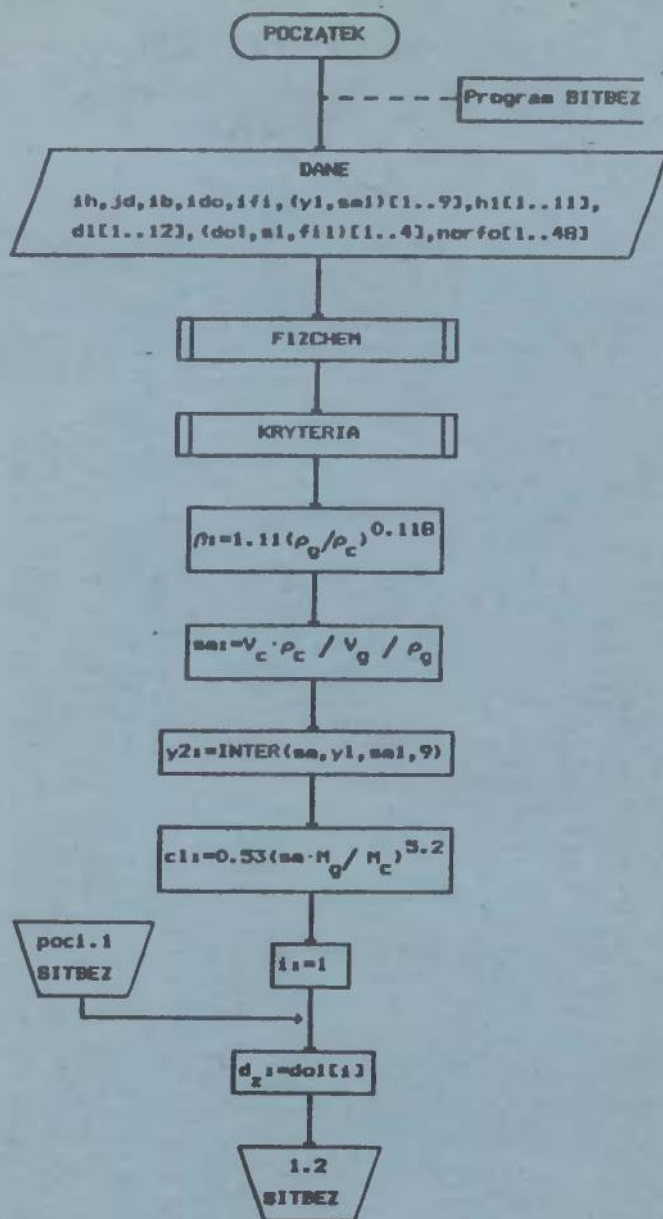


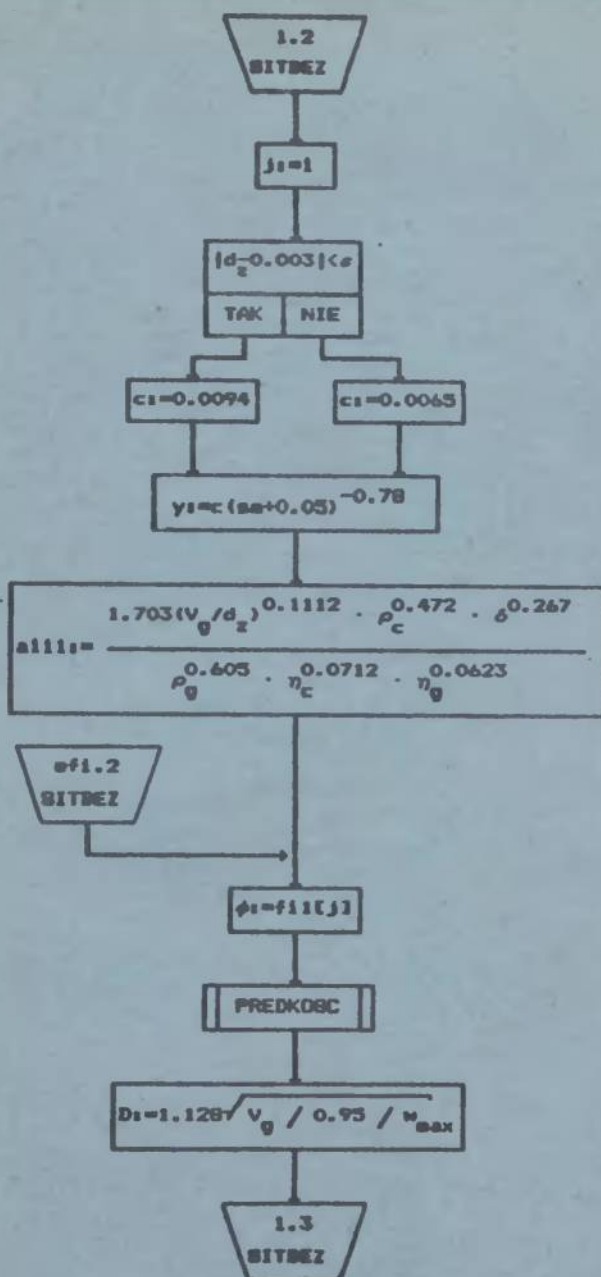




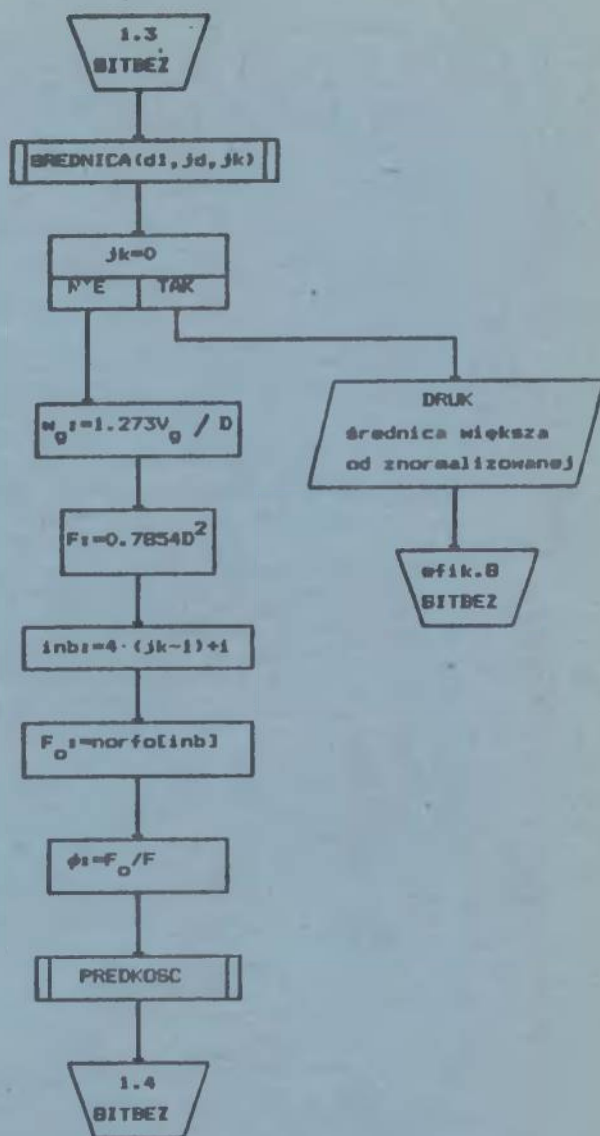


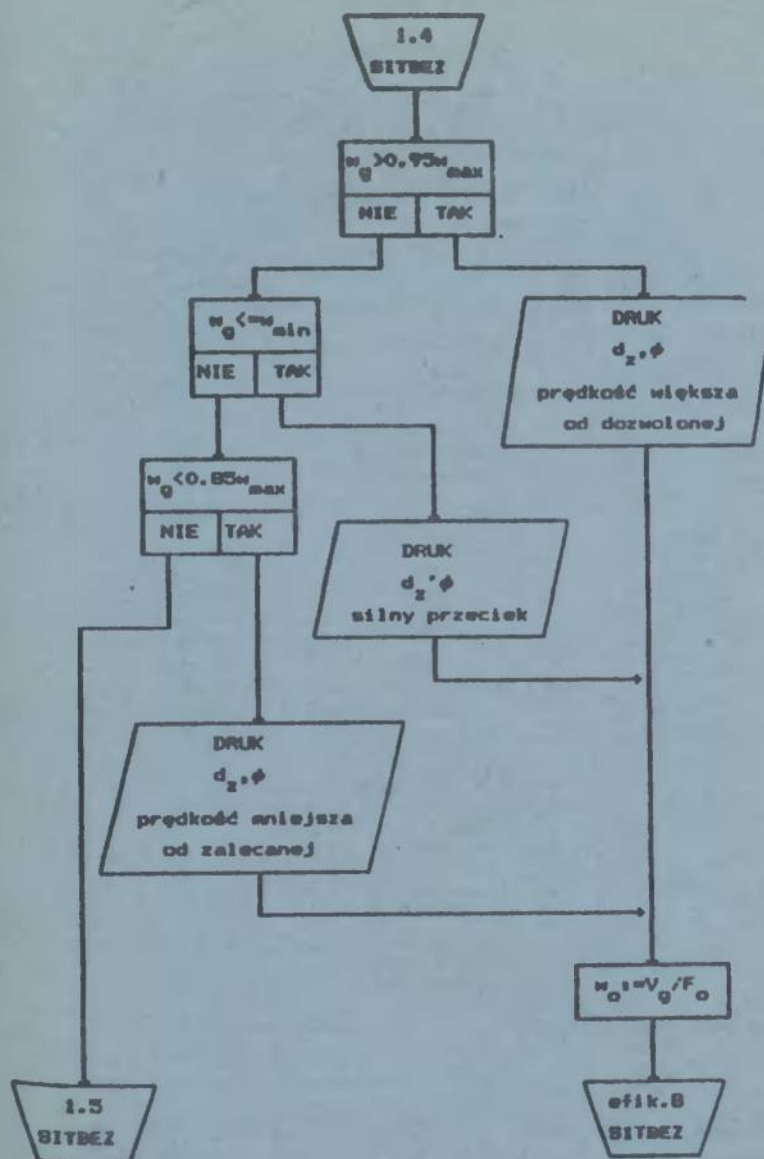
5.4. Schemat blokowy programu SITBEZ

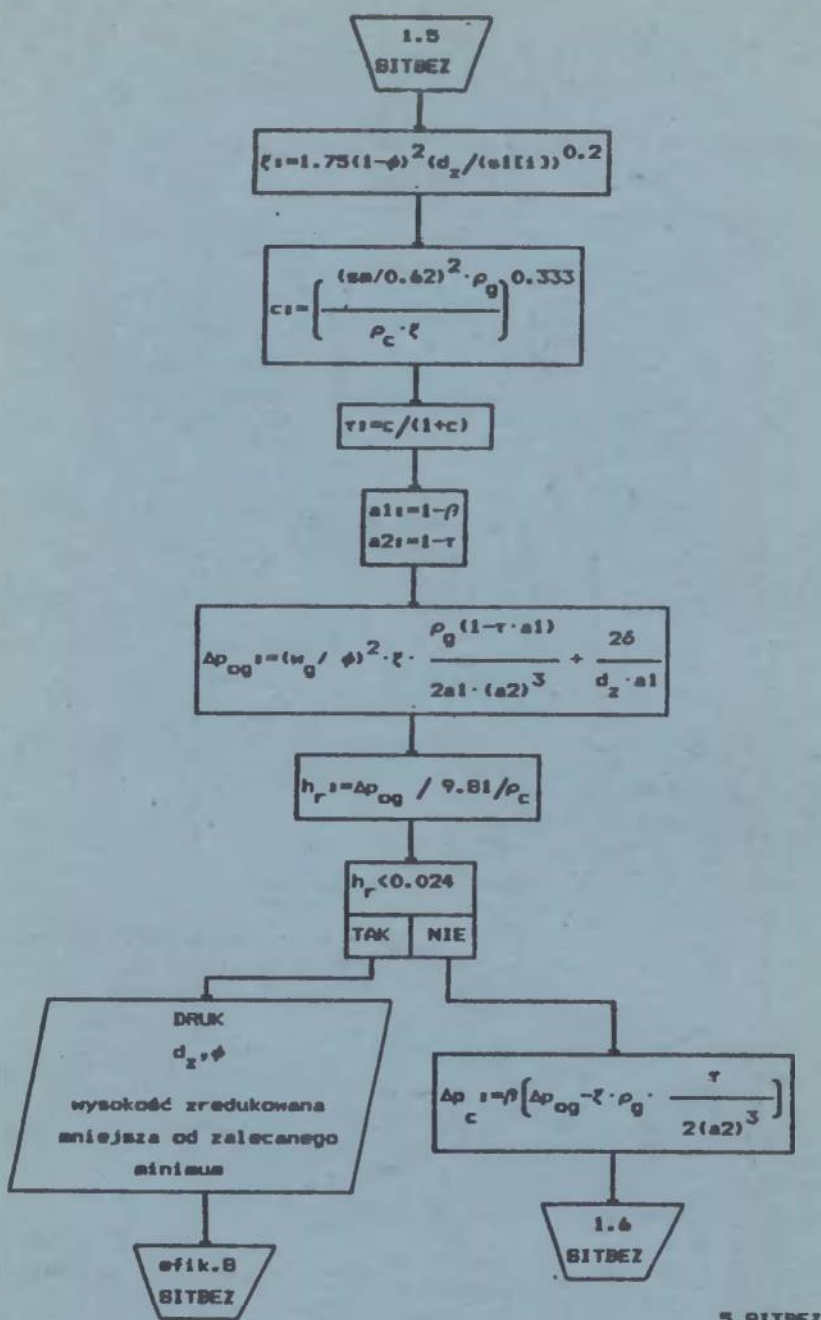


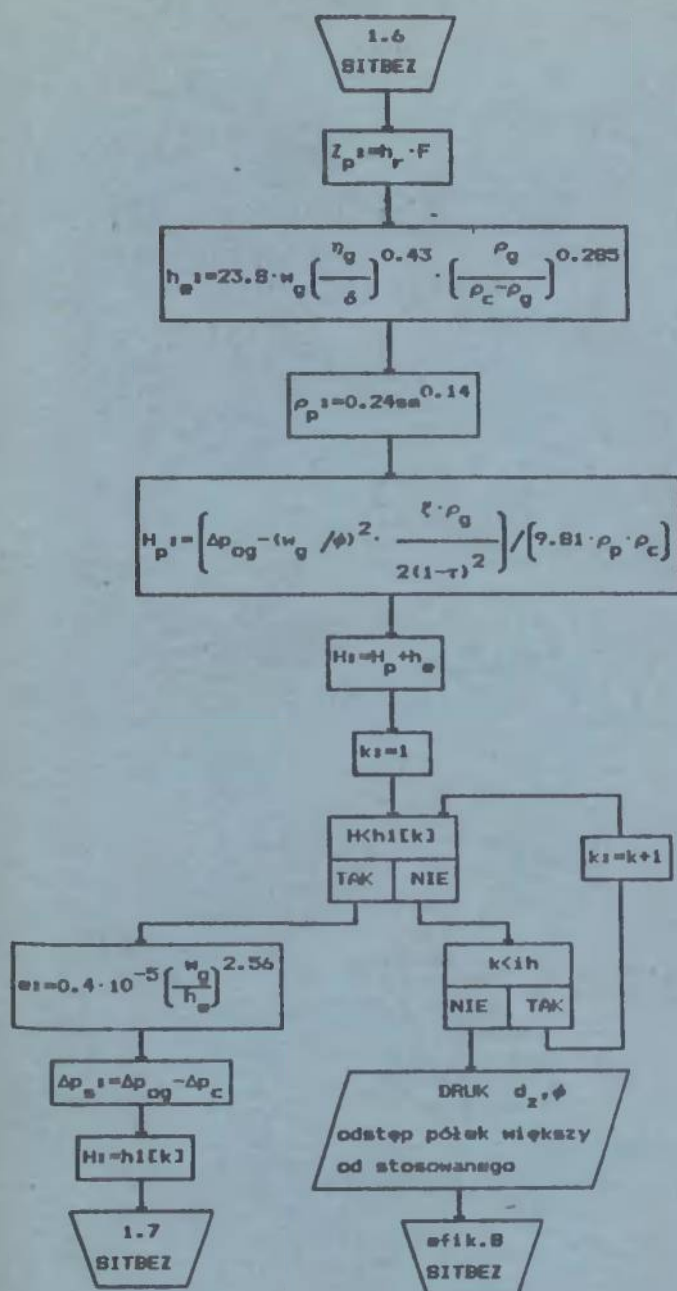


2.SITBEZ

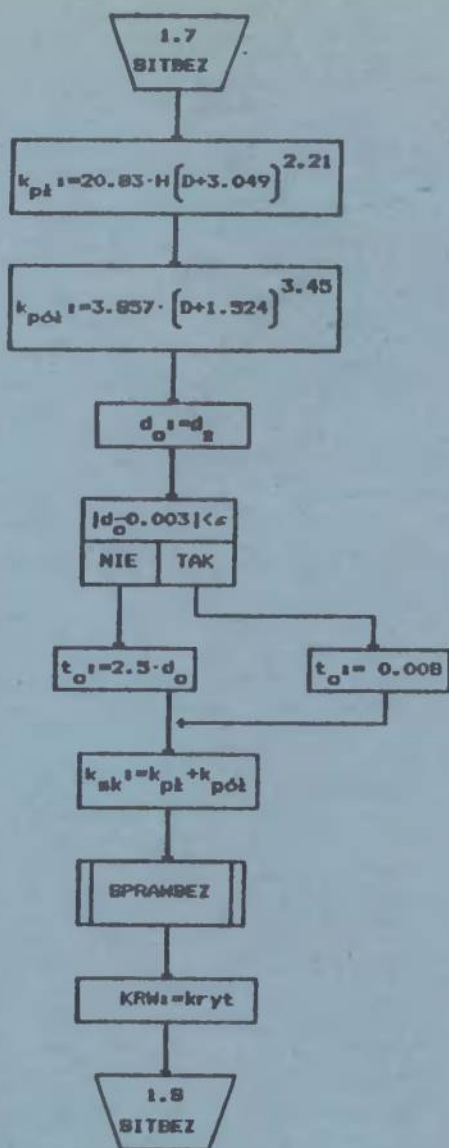


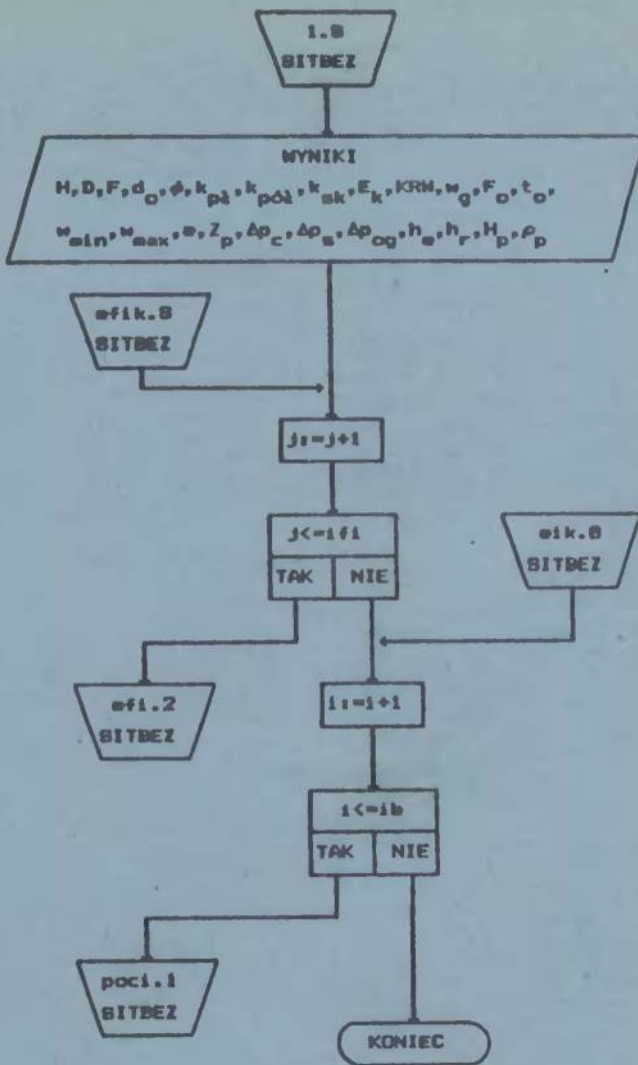




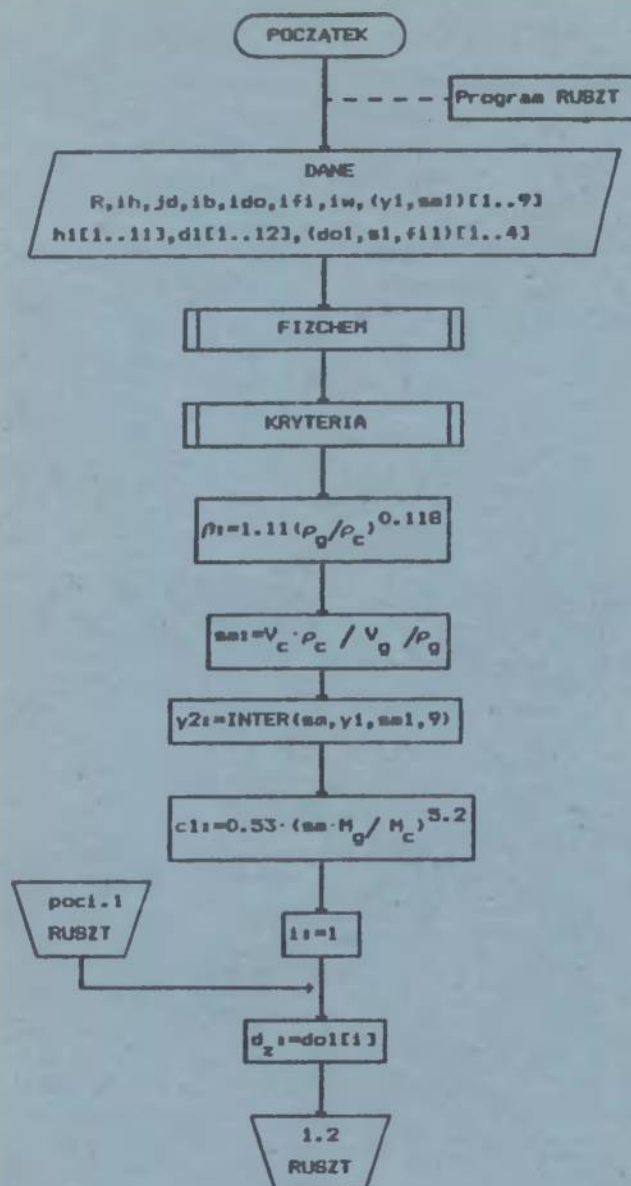


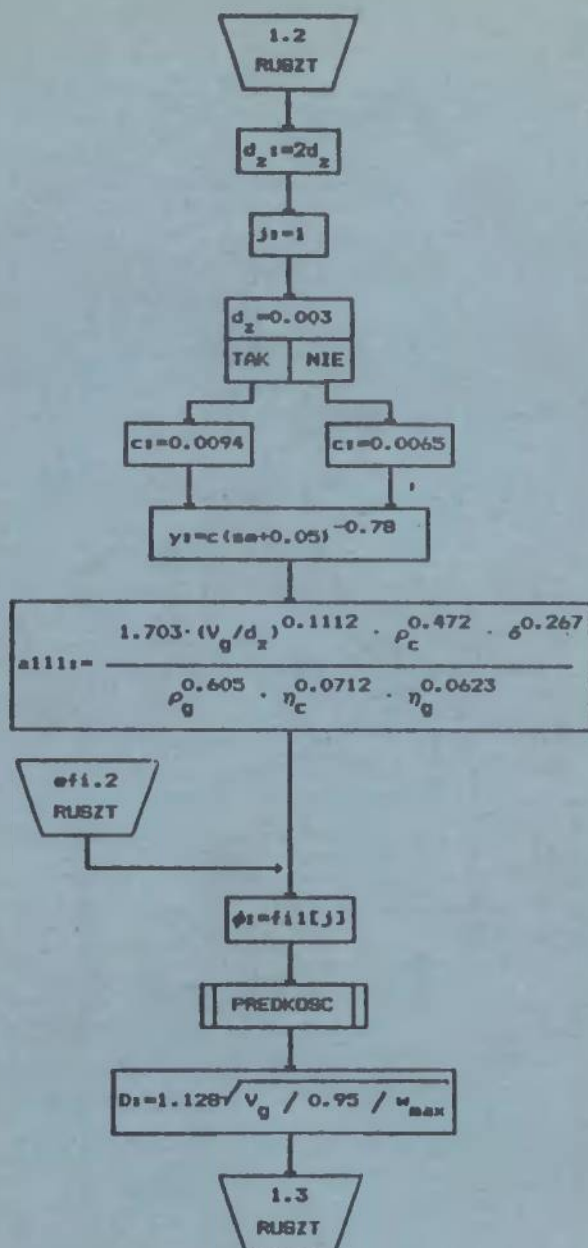
6. BITBEZ

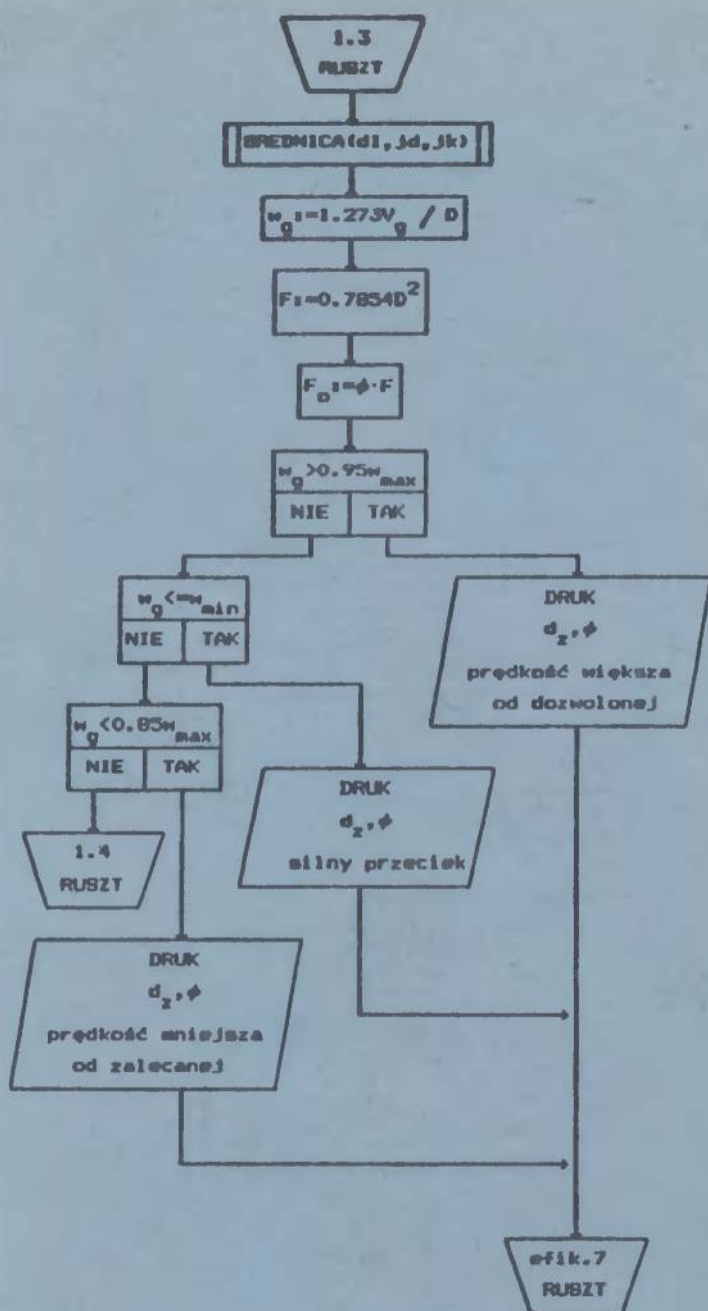


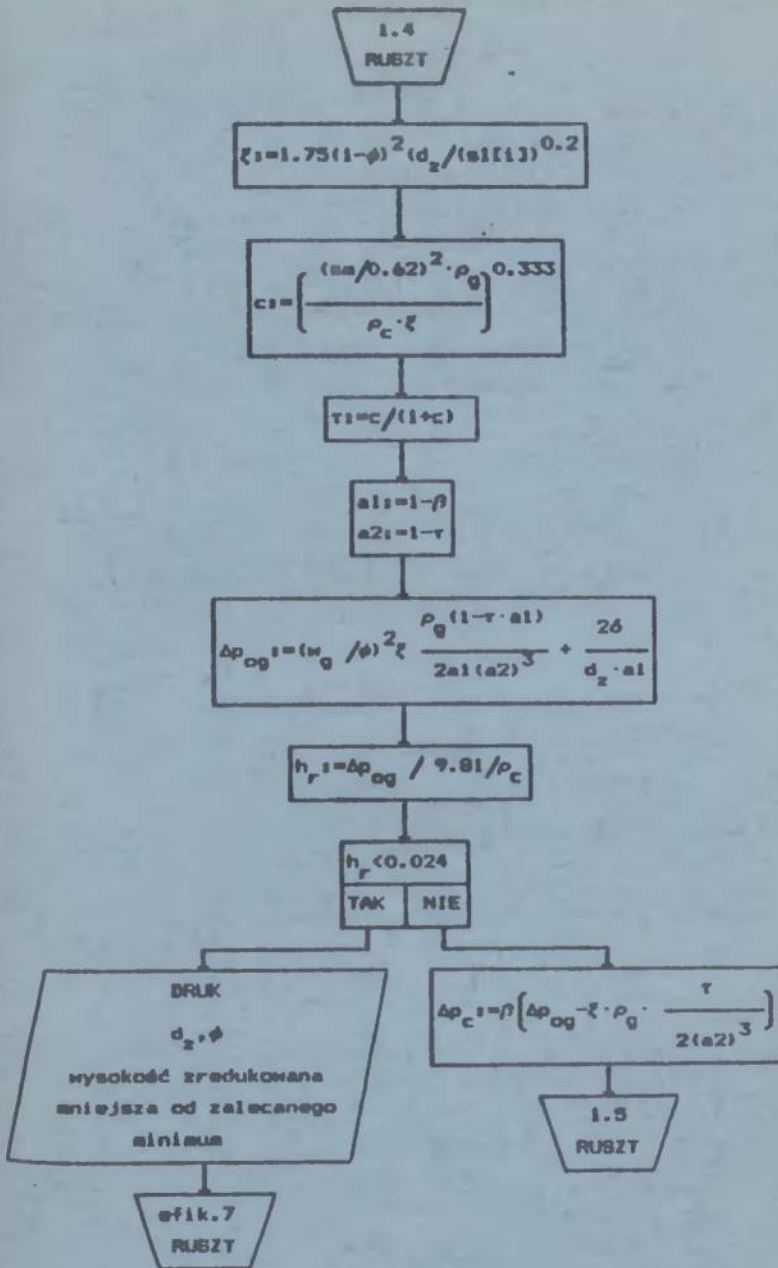


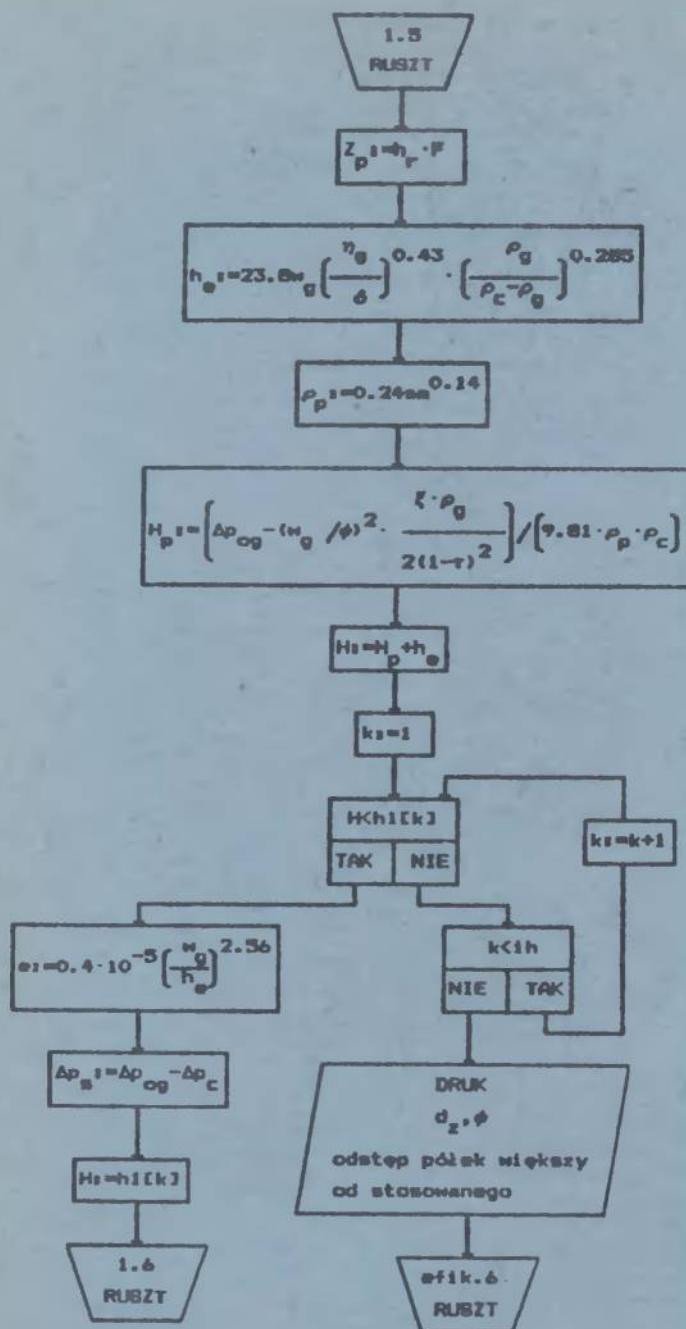
5.5. Schemat blokowy programu RUBZT

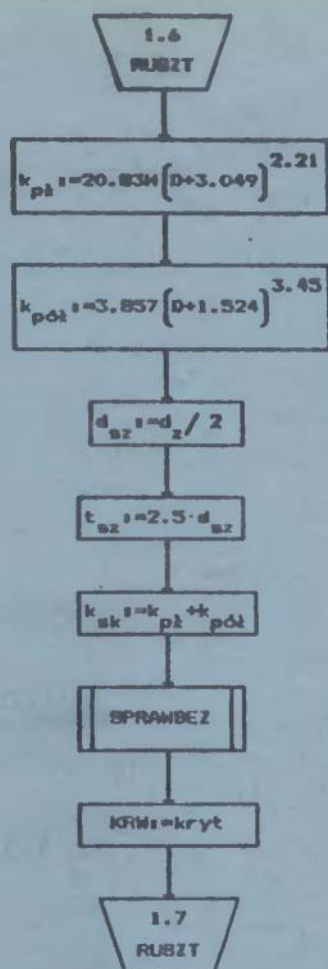


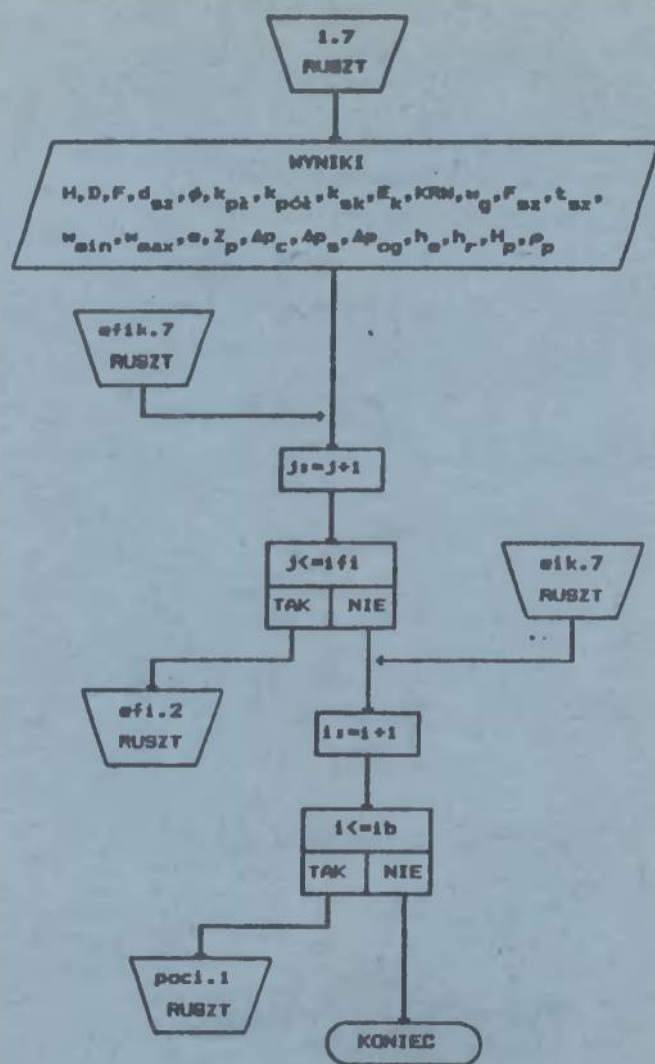












6. TABULOGRAMY PROGRAMÓW

6.1. Tabulogram programu KOLP

```

program KOLP(dane,dane1,dane2,dane3,dankolp,tabkolp,tabwar,wynkolp,
             wybkolp);
uses CRT,printer;

var dankolp,dane,dane1,dane2,dane3,tabkolp,tabwar,wynkolp,
    wybkolp:text;

label pocz1,poci,eil,314,214,edr,eca,edri,ee,ehp,ehpk,ewyb;
var a,ap,i,ih,ii,ik,ini,io,ir,iw,j,jd,jn,jo,jk,jlp,k,ka,kaf,kd,
    ki,kkr,krok,ks,a,as,nf,nlp,nlsk,ncsk,ns,u:integer;
    a1,a2,a3,af,ag1,ag3,alfa,alfas,as,b,baa,bi,bp,c,ca,csk,cg,d,
    dd,des,dk,dr,dyc,e,emp,ewp,etc,etg,etcs,etgs,ek,wlok,f,fip,
    faf,fo,fw,fr,fp,fp1,fi,fa,fwg,h,hg,hw,hk,hrn,hwi,hs,hap,hp,
    hp2,hpk,hwp,hr,hgi,hwc,hpc,hpe,hz,hpi,he,hc,hwr,is,kru,kre,
    kgy,kgr,kk,kp,kpo,kak,l,li,lcc,lq,lqc,lde,lbp,lip,lp,lak,me,
    nig,nc,ag,eca,ags,axy,nsi,nw,pt,pg,pcal,pog,pr,r,ri,rv,roc,
    rog,roca,roga,s,ap,si,sis,samak,su,t,tg,tc,tdc,tdg,tgc,tgg,tp,
    t0,ts,tw,ta,ui,u2,u3,wa,w,wyb,v,vg,vc,vdg,vgg,vmol,x,xg,
    xpri,y,ye,zp:real;
norma,norri:array[1..37]of integer;
norrozi:array[1..37]of char;
bila,dan,dana,reak,rekt,roz,jed,znak:char;
rek:string[20];dak:string[16];

const wil=14;
    wve=5;
    wiw=20;
    eps1 = 1e-6;

type
    wektor=array[1..wil] of real;
    vector=array[1..wil] of real;
    vect=array[1..wve] of real;
    tablica=array[1..wil,1..wiw] of real;
    tabel=array[1..wil,1..wve] of real;

```



```

    tabl1=array[1..w1,1..w1] of real;
    vecto=array[1..24] of real;
    tabs=array[1..4,1..14] of real;
var  aa,ba,ca,c2,v1,y2,h1,e1,h2,y1,x1,t1,c3,b3,fgr,b2,ro,xp,yp,aa,
    pch,rch,tkr,pkr,zd,xz,xw,xx,xsur,kr,ks,yd,ys,yw:wektor;
    tsu,tf,fs:vekt;
    cel,e1:tabs;
    eo,xsr,xft:tabel;
    rc,rg,hpl,hk1:tablica;
    ww:tabl1;
    di:vekt;

($I as \polki\podprogr\potega.pas)
($I as \polki\podprogr\intablic.pas)
($I as \polki\podprogr\inter2.pas)
($I as \polki\podprogr\interpol.pas)
($I as \polki\podprogr\przelew.pas)
($I as \polki\podprogr\wybered.pas)
($I as \polki\podprogr\sprawno.pas)
($I as \polki\podprogr\fizchem.pas)
($I as \polki\podprogr\krytyk.pas)
($I as \polki\podprogr\kryteria.pas)

procedure gradcis;
var  lin,igi:integer;
    ag2,bg3,cg3,fg,f1,f11,g1,hgrp:real;
    label elin,eig,egrad,eqr;
begin
    ag2:=118.181r*(hp+hs+hpk*(dk-dr)*(1k/1r+1)/((t-dk)*1k/1r+0.15));
    lin:=0;
    elin:=1;
eig:  f1:=pot(hg,1.5)*ag1+sqrt(hg)*ag2-ag3;
    f11:=1.5*sqrt(hg)*ag1+0.5*ag2/sqrt(hg);
    hgi:=hg-f1/f11;
    if abs(hgi-hg)<=0.005 then begin hg:=(hg+hgi)/2; goto eqr;end;

```

```

if ig<20 then begin ig:=ig+1;hg:=hg1;goto e1g;end;
hg:=hg+0.1;
if (hg<1.31) and (lin<20) then begin lini:=lin+1;goto e1in end;
writeln('H=',h,'D=',d,'hp=',hp,'hg=',hg,'f1=',f1,
        'algorytm newtona rozbieżny');halt;
egr: fg:=0.819738fmg;
if (fg<0.4) or (fg>1.4) then
begin
    writeln('argument poprawki',
            'gradientu poza przedzia',
            'lem interpolacji');
    goto egrad;
end;
if fg<>1.1 then begin
    gl:=10.76481bp;
    if gl<0.167 then
begin
        cg3:=inter(fg,c3,fg,12);
        bg3:=inter(fg,b3,fg,12);
        hgrp:=bg3*gl+cg3;
        hg:=hg+hgrp;
    end;
end;
egrad: hg:=hg+0.0234;
end;(obliczania gradientu cisnienia)
begin
    ClrScr;
    assign(dankolp,'a:\polki\zbiordan\dankolp.pas');
    reset(dankolp);
    readln(dankolp,pcal,jd,ih,m,as);
    assign(tabkolp,'a:\polki\zbiordan\tabkolp.pas');
    reset(tabkolp);
begin

```

```

for i:=1 to 6 do read(tabkolp,c1[i]);
for i:=1 to 6 do read(tabkolp,h2[i]);
for i:=1 to 6 do read(tabkolp,y1[i]);
for i:=1 to 14 do read(tabkolp,c2[i]);
for i:=1 to 14 do read(tabkolp,v1[i]);
for i:=1 to 14 do read(tabkolp,y2[i]);
for i:=1 to 7 do read(tabkolp,x1[i]);
for i:=1 to 4 do read(tabkolp,b2[i]);
for i:=1 to 6 do
    for j:=1 to 7 do read(tabkolp,cs1[i,j]);
for i:=1 to 4 do
    for j:=1 to 14 do read(tabkolp,e1[i,j]);
for i:=1 to jd do read(tabkolp,d1[i]);
for i:=1 to ih do read(tabkolp,h1[i]);
for i:=1 to 3 do
    for j:=1 to s do read(tabkolp,hp1[i,j]);
for i:=1 to 3 do
    for j:=1 to s do read(tabkolp,hk1[i,j]);
for i:=1 to 12 do read(tabkolp,fgf[i]);
for i:=1 to 12 do read(tabkolp,b3[i]);
for i:=1 to 12 do read(tabkolp,c3[i]);
for i:=1 to 37 do read(tabkolp,norea[i]);
for i:=1 to 37 do read(tabkolp,norir[i]);
for i:=1 to 37 do read(tabkolp,norroz[i]);

endj
fizche;
kryteria;
pocz1:=1;wyb:=1;
assign(wybkolp,'c:\polki\wybkolp.pas');
rewrite(wybkolp);
if (dan='t') or (bila='t') then
begin

```

```

writeln(wybkolp, '');
writeln(wybkolp, '          WYNIKI OBLICZEN POLEK',
        ' KOLPAKOWYCH');
writeln(wybkolp, '          dla zalozonych parametrow',
        ' ruchowych (Vg, Vc)');
writeln(wybkolp, '
');
writeln(wybkolp, '          Vg=, vg:6:4, ' m3/s',
        '          Vc=, vc:7:5, ' m3/s');
writeln(wybkolp, '
');
writeln(wybkolp, ' ');
writeln(wybkolp, ' ');
end
else
begin
writeln(wybkolp, '');
writeln(wybkolp, '          WYNIKI OBLICZEN POLEK',
        ' KOLPAKOWYCH');
writeln(wybkolp, '          dla zalozonych parametrow',
        ' ruchowych (Vg, Vc, R)');
writeln(wybkolp, '
');
writeln(wybkolp, '          Vg=, vg:6:4, ' m3/s',
        '          Vc=, vc:7:5, ' m3/s', '          R=, r:4:1);
writeln(wybkolp, '          ', r:4:1);
writeln(wybkolp, ' ');
writeln(wybkolp, ' ');
end;
assign(wynkolp, 'c:\polki\wynkolp.pas');
rewrite(wynkolp);
if (dan='t') or (bila='t') then
begin
writeln(wynkolp, '');
writeln(wynkolp, ' ');
writeln(wynkolp, '          WYNIKI OBLICZEN POLEK',

```

```

        ' KOLPAKOWYCH');
writeln(wynkolp, '          dla zalozonych parametrow',
        ' ruchowych (Vg,Vc)');
writeln(wynkolp, ' ');
writeln(wynkolp, '          Vg=',vg:6:4,' m3/s',
        '          Vc=',vc:7:5,' m3/s');
writeln(wynkolp, ' ');
writeln(wynkolp, ' ');
end
else
begin
    writeln(wynkolp, '');
    writeln(wynkolp, ' ');
    writeln(wynkolp, '          WYNIKI OBLICZEN POLEK',
        ' KOLPAKOWYCH');
    writeln(wynkolp, '          dla zalozonych parametrow',
        ' ruchowych (Vg,Vc,R)');
    writeln(wynkolp, ' ');
    writeln(wynkolp, '          Vg=',vg:6:4,' m3/s',
        '          Vc=',vc:7:5,' m3/s', '          R=',r:4:1);
    writeln(wynkolp, '          ',rek);
    writeln(wynkolp, ' ');
end;
poc1:=h1;
c:=inter(h,c1,h2,6);
if ini =0 then
begin
    writeln('Parametr c poza przedzialem interpolacji');
    writeln('SPRAWDZ DANE');
    halt
end;
w1=9.47e-5c:8sqrt((roc-rog)/rog);
d1=1.128sqrt(vg/w);

```



```

        bi:=0.01; bma:=0.023; kk:=0.4833; ki:=2
    end;
3: begin dk:=0.08; nws=26; tw:=0.00966; hw:=0.025; fw:=0.00311;
        fr:=0.00221; dr:=0.057; hk:=0.065; hrn:=0.06; fpi:=0.00247;
        bi:=0.01; bma:=0.022; kk:=0.5417; ki:=1;
    end;
end; (case kd)
eil:=sqrt(dsd-lp*lp)/2;
if jk<=2 then jn:=(jk-1)*2+jlp-1 else jn:=(jk-1)*3+jlp-2;
ik:=norma[jn]; ir:=norir[jn]; roz:=norroz[jn];
a1:=vg*hw/ik/fw; a2:=rog/(roc-rog);
hw1:=0.755*pot(a1,0.6667)*pot(a2,0.3333);
if (hw1/hw)<0.9 then begin
    przel(dk,t,lp,nlp,jlp);
    if nlp=0 then goto eil
    else
        begin
            writeln(wynkolp,'***H=',
                h:4:2,' D=',d:4:2,
                ' hp=',hp:6:4,
                ' *** kolpaki nie',
                ' wykorzystane');
            goto 314
        end
    end
else if hw1/hw>1.1 then
    begin
        writeln(wynkolp,'***H=',
            h:4:2,' D=',d:4:2,
            ' hp=',hp:6:4,'***',
            ' za duze obciazenie',
            ' gazem');
        goto 314
    end

```

```

end;

a1:=vg/ik/fr;
pt:=0.2748kk$a1$a1$a2;
b:=lp/d;
a1:=sqrt(1-b*b);
fp:=dsds(arctan(b/a1)-b$a1)/2;
bp:=(f-fp)/2/s;
y:=0.828wsqrt(rpg);
x:=4824$vc/bp;
if x>64 then begin przel(dk,t,lp,nlp,jlp);
                    if nlp=0 then goto eil else goto edr;
                end;
if (x<5) or (y<0.6) or (y>1.4) then goto edr else goto wcs;
edr:writeln(wynkolp,'$$$H=',h:4:2,' D=',d:4:2,' hp=',hp:5:3,
            ' $$$ parametry cs poza przedzialem interpolacji');
goto 314;
wcs:cs:=indwa(x,y,x1,y1,cs1,7,6);
lbp:=vc/bp;
xg:=3600$lbp;
if (xg<=2.5) or (xg>=100) then
begin
    writeln(wynkolp,'$$$H=',h:4:2,' ',
            'D=', d:4:2,' hp=',hp:5:3,
            ' $$$ parametry cg poza',
            ' przedzialem interpolacji');

    goto 314
end;

cg:=inter(xg,c2,v1,14);
ag1:=1.5$lr-1.4;
ag3:=187.7482$vc$pot(lr,1.5)/(cg$(1k$(t-dk)/lr+0.15));
ye:=813.6$vc/pot(lp,2.5);
if ye>30 then

```

```

begin przel(dk,t,lp,nlp,jlp);
  if nlp=0 then goto e11 else goto edr1;
end;

if ye>0.2 then goto ee;
edr1:writeln(wynkolp,'$$$H=',h:4:2,' D=',d:4:2,' hp=',hp:5:3,' ye=',
  ye:4:4,'$$$$$$$$$parametr e poza przedzialem',
  'interpolacji');

goto 314;

ee:ee:=indxa(ye,b,y2,b2,e1,14,4);
a2:=vc/lp;
hs:=0.667*espot(a2,0.667);
if(2*ss)>3.0 then ka:=1 else if(2*ss)<1.5 then ka:=2 else ka:=3;
case ka of
  1:hp1:=0.0375;
  2:hp1:=0.0125;
  3:hp1:=0.025;
end;(case ka)
ehp:hp:=hp1[k1,k];
hp2:=1.5*hs+hp;
if hp<hp2 then
  begin
    writeln(wynkolp,'$$$H=',h:4:2,' D=',d:4:2,
      ' hp=',hp:5:3,'$$$za mala wysokosc przelewu');
    goto 214;
  end;

ka:=k;
ehpk:hpka:=hk1[k1,ka];
r1:=hpka+hw+0.005;
hwp:=hp-r1;
if hwp<0.005 then
  if (ka=1) or (ap=1) then
    begin
      writeln(wynkolp,'$$$H=',

```

```

        h:4:2,' D=',d:4:2,
        ' hp=',hp:5:3,'$$$za',
        'male pokrycie wyciec',
        ' przy nieruchomej ',
        'cieczy.zwiekszye hp');
    goto 2i4;
end
else
begin
    ka:=ka-1;ap:=0;goto ehpk
end;

if hwp>=0.03 then
    if (ka=m) or (ap=0) then
        begin
            writeln(wynkolp,'$$$H=',
                h:4:2,' D=',d:4:2,
                ' hp=',hp:5:3,
                '$$$za duze',
                ' pokrycie wyciec',
                ' przy nieruchomej',
                ' cieczy. ');
            goto 3i4
        end else
        begin
            ka:=ka+1;ap:=1;goto ehpk
        end;

    if (hrn-r1)<bi then hr:=r1+bi
        else if hrn<=(r1+bma) then hr:=hrn
            else hr:=r1+bma;

    hg:=0.05;
    gradcis;
    hg:=hg*1;
    hwc:=hs+0.5*hg+hwp;

```



```
if hwc<0.025 then
```

```
  if (ka=1) or (ap=1) then
```

```
    begin
```

```
      writeln(wynkolp,'$$$H=',
```

```
        h:4:2,' D=',
```

```
        d:4:2,' hp=',
```

```
        hp:5:3,'$$$za',
```

```
        ' male pokrycie',
```

```
        ' wyciec przy',
```

```
        ' przeplywie',
```

```
        ' cieczy');
```

```
      goto 214;
```

```
    end
```

```
  else
```

```
    begin
```

```
      ka=ka-1;ap:=0;goto ehpk
```

```
    end;
```

```
if hwc>0.05 then
```

```
  begin
```

```
    if (ka=m) or (ap=0) then
```

```
      begin
```

```
        writeln(wynkolp,'$$$H='
```

```
          ,h:4:2,' D=',
```

```
          d:4:2,' hp=',
```

```
          hp:5:3,'$$$za',
```

```
          ' duzy poziom',
```

```
          ' hwc');
```

```
        goto 214 ;
```

```
      end
```

```
    else begin
```

```
      ka=ka+1;ap:=1;
```

```
      goto ehpk
```

```
    end;
```

```

end;

pg:=pt+hw1;
rv:=hg/pg;
if rv>0.5 then
begin
  writeln(wynkolp,'sssH=',h:4:2,' D=',d:4:2,
    ' hp=', hp:5:3,
    ' $$$polka nie stabilna');
  goto 214
end;

pog:=pg+hs+hwp+hg/2;
hpp:=hp-hsp;
pr:=vc/lp/hpp;
pr:=0.1659*pr*pr;
hpe:=hp+hs+pr+pog+hg;
if hpe>(h/2) then
begin
  writeln(wynkolp,'sssH=',h:4:2,' D=',d:4:2,
    ' hp=', hp:5:3,
    ' $$$za duzy poziom',
    ' cieczy w elemencie splywowym');
  goto 214
end;

ta:=0.8*sqrt(hs*(h+hp-hpe));
if ta>(0.68*(d/2-s)) then
begin
  writeln(wynkolp,'sssH=',h:4:2,' D=',
    d:4:2, ' hp=',hp:5:3,' $$$nie',
    ' rowny przeplyw cieczy');
  goto 214
end;

fa:=f-fp;
for:=1*sfpr;

```

```

wa:=vg/fa;
hz:=hwc+hp;
hpi:=0.043185*vg*vg*rog/fa/fa+1.07*hp-0.04064;
if hpi>=h then
    begin
        writeln(wynkolp,'***H=',h:4:2,'    D=',d:4:2,
            '    hp=',hp:5:3,'***za male ',
            'odleglosc miedzy polkami');
        goto 314
    end;
he:=h-hpi;
a2:=vg/fa/he;
ae:=4.69e-6*pot(a2,2.31)/si;
hc:=0.17*hp-0.0135*vg*sqrt(rog)/fa+2.4532*vc/bp+0.0409;
zp:=hc*(fa-fo)+hp*fp/2;
hwr:=hr-r1;
sprawnos;
kp:=20.83*hp*pot((d+3.049),2.21);
kpo:=5.1234*pot((d+1.524),3.45);
ksk:=kp+kpo;
krw:=kryt;
vi:=9.81*roc;
pt:=pt*v;
pr:=pr*v;
pg:=pg*v;
pog:=pog*v;
fp:=fp/2;
writeln(wynkolp,'');
writeln(wynkolp,'*** H=',h:4:2,' m', '    D=',d:4:2,' m',
    '    F=',f:6:4,'          wg=',w:6:4,' m/s');
writeln(wynkolp,'');
writeln(wynkolp,'          KPD=',kpo:5:2,'          KP=',kp:6:2,
    '          KSK=',ksk:6:2);

```

```

writeln(wynkolp,"          dk=","dk:6:4,
          "          KOW=","kwi:7:2);
writeln(wynkolp,"");
writeln(wynkolp," 88 hp=","hp:5:3,"a","          lp=","lp:5:3,"a");
writeln(wynkolp,"");
writeln(wynkolp,"          t          dk          fa",
          "          fo          ha");
writeln(wynkolp,"          a          a          a2",
          "          a2          a");
writeln(wynkolp,t:11:3,dk:14:3,fa:13:4,fo:14:4,ha:13:4);
writeln(wynkolp,"          s          bp          fp",
          "          hps          hpp");
writeln(wynkolp,"          a          a          a2",
          "          a          a");
writeln(wynkolp,s:11:4,bp:14:4,fp:13:4,hps:14:4,hpp:13:4);
writeln(wynkolp,"          ik          ir          roz",
          "          ns          hr");
writeln(wynkolp,"          szt          szt ",
          "          a");
writeln(wynkolp,iks:11,irs:14,roz:13,ns:14,hrs:13:4);
writeln(wynkolp,"          hwr          hpk          hwp",
          "          hwc          hw1");
writeln(wynkolp,"          a          a          a",
          "          a          a");
writeln(wynkolp,hws:11:4,hps:14:4,hws:13:4,hws:14:4,hws:13:4);
writeln(wynkolp,"          pr          pg          pt",
          "          hg          pog");
writeln(wynkolp,"          N/a2          N/a2          N/a2",
          "          a          N/a2");
writeln(wynkolp,pr:11:2,pg:14:2,pt:13:2,hg:14:4,pog:13:2);
writeln(wynkolp,"          ta          ae          hpi",
          "          hc          zp");
writeln(wynkolp,"          a          k6/k6          a",

```

```

                                m          m3');
writeln(wynkolp,tai1:4,me1:4,mpi:13:4,hc1:14:4,xpi13:4);
writeln (wynkolp,'#####',
          '#####');
writeln(wynkolp,'');
if (wyb=1) or (wyb=krw) then
  append(wybkolp) else if (wyb>krw) then
    begin
      assign(wybkolp,'c:\polki\wybkolp.pas');
      rewrite(wybkolp);
      if (dan='t') or (bila='t') then
        begin
          writeln(wybkolp, '');
          writeln(wybkolp,'          WYNIKI OBLICZEN POLEK',
                    ' KOLPAKOWYCH');
          writeln(wybkolp,'          dla zalozonych parametrow',
                    ' ruchomych (Vg,Vc)');
          writeln(wybkolp, '');
          writeln(wybkolp,'          Vg=',vg:6:4,' m3/s',
                    '          Vc=',vc:7:5,' m3/s');
          writeln(wybkolp,'          ');
          writeln(wybkolp, '');
        end
      else
        begin
          writeln(wybkolp, '');
          writeln(wybkolp,'          WYNIKI OBLICZEN POLEK',
                    ' KOLPAKOWYCH');
          writeln(wybkolp,'          dla zalozonych parametrow',
                    ' ruchomych (Vg,Vc,R)');
          writeln(wybkolp, '');
          writeln(wybkolp,'          Vg=',vg:6:4,' m3/s',
                    '          Vc=',vc:7:5,' m3/s', '          R=',r:4:1);

```



```

        writeln(wybkolp, '                                ', rak);
        writeln(wybkolp, ' ');
    end;

    end else goto swyb;

    writeln(wybkolp, '');
    writeln(wybkolp, ' $$$ H= ', h:4:2, ' m', '          D= ', d:4:2, ' m',
        '          F= ', f:6:4, '          wg= ', w:6:4, ' e/m');
    writeln(wybkolp, '');
    writeln(wybkolp, '          KPO= ', kpo:5:2, '          KP= ', kp:6:2,
        '          KSK= ', ksk:6:2);
    writeln(wybkolp, '          Ek= ', ek:6:4,
        '          KRW= ', krw:7:2);
    writeln(wybkolp, '');
    writeln(wybkolp, ' $$ hp= ', hp:5:3, ' m', '          lp= ', lp:5:3, ' m');
    writeln(wybkolp, '');
    writeln(wybkolp, '          t          dk          fa',
        '          fo          ha');
    writeln(wybkolp, '          m          m          m2',
        '          m2          m');
    writeln(wybkolp, t:11:3, dk:14:3, fa:13:4, fo:14:4, ha:13:4);
    writeln(wybkolp, '          S          bp          fp',
        '          hpe          hpp');
    writeln(wybkolp, '          m          m          m2',
        '          m          m');
    writeln(wybkolp, s:11:4, bp:14:4, fp:13:4, hpe:14:4, hpp:13:4);
    writeln(wybkolp, '          ik          ir          roz',
        '          ns          hr');
    writeln(wybkolp, '          szt          szt ',
        '          m');
    writeln(wybkolp, ik:11, ir:14, roz:13, ns:14, hr:13:4);
    writeln(wybkolp, '          hwr          hpk          hwp',
        '          hwc          hw1');
    writeln(wybkolp, '          m          m          m',

```

```

      '      m      m' );
writeln(wybkolp,hwr:11:4,hpk:14:4,hwp:13:4,hwc:14:4,hwl:13:4);
writeln(wybkolp,'      pr      pg      pt',
      '      hg      pog');
writeln(wybkolp,'      N/m2      N/m2      N/m2',
      '      m      N/m2');
writeln(wybkolp,pr:11:2,pg:14:2,pt:13:2,hg:14:4,pog:13:2);
writeln(wybkolp,'      ta      ae      hpi',
      '      hc      zp');
writeln(wybkolp,'      m      k6/k6      m',
      '      m      m3');
writeln(wybkolp,ta:11:4,ae:14:4,hpi:13:4,hc:14:4,zp:13:4);
writeln(wybkolp,'!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!',
      '!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!');
writeln(wybkolp,'');
wyb:if (wyb=1) or (wyb>krw) then wyb=krw;
214:if k<m then begin k:=k+1; goto ahpjend;
314:if i<ih then begin i:=i+1; goto paci end;
      close(wynkolp);close(wybkolp);
end. ( koniec programu KOLP )

```

6.2. Tabulogram programu ZANOR

```

program ZANOR(dane,dane1,dane2,dane3,danzawor,databzaw,tabwar,
              wynzawor,wybzawor);

uses crt,printer;

var dane,dane1,dane2,dane3,danzawor,databzaw,tabwar,wynzawor,
    wybzawor:text;

label poczi,poc1,efka,ed1,ej,et,ebp,eibn,ekon,ewo,ebn,enz,
    ebeg,edr1,eap,eye,ehp,eifw,ez,epc1,eps,apc,epog,eha,en2,ewyb;

var az,bn,i,ii,id,ih,ik,ini,ip,iw,il,i2,j,k,kkr,es,n,nf,ni,ns,
    ncsk,nlsk,tz,ti,u,q:integer;

a,ad,a7,alfa,alfan,ap,as,a1,a2,a3,a4,b,bp,cak,d,dd,des,de,dor,
dyc,s,ek,elok,ewg,ewp,etc,etcs,etg,etgs,f,fa,fan,ff,fip,fka,fa,
faf,fel,fo,fp,fpb,fps,fp1,fp2,fv,fz,f1,gz1,gz2,h,hc,hp,hpa,hpi,
hpp,hs,hsp,kl,k2,k3,k4,kgr,kgv,kp,kpo,kre,krw,ka,kak,l,l1,lc,
lcc,lq,lqc,lqc,lp,lp11,lp2,lsk,ms,mc,mca,ag,ags,aig,axy,ez,az1,
az2,nal,nt,nz,n1,n2,pc,pcal,pog,pr,ps,r,roc,roca,rog,roga,rop,
s,si,sia,sp,sp1,sp2,spe,suak,su,t,ta,tg,tc,tdc,tdg,tgc,tgg,tp,
ts,t0,u1,u2,u3,v,vml,vc,vdg,vvg,vg,vo,w,wa,wc,wg,wol,wo2,wp,
wr,w1,w2,w3,wyb,x,xp1,y,ye,z,xp:real;

hpl:array [1..7] of real;
lpl:array [1..4] of real;
ms,gs:array [1..4] of real;
nb:array [1..18,1..4,1..10] of real;
nbl:array [1..16,1..4,1..12] of real;
hl:array [1..11] of real;

bila,dana,jed,znak,dan,reak,rekt:char;
rek:string[20];
dak:string[16];
typz:string[16];
natrum:string[23];

const ia=3;

dot=0.039;dz=0.048;g=0.005;na=3;am1=0.003;am=0.023;am1=0.018;

```

```

    wil=14; wve=5; wiw=20;
    epsi=1e-6;
type wektor=array[1..wil] of real;
    tabe=array [1..5,1..14] of real;
    tablica=array [1..wil,1..wiw] of real;
    tabel=array[1..wil,1..wve] of real;
    vect=array[1..wve] of real;
    tab11=array[1..wil,1..wil] of real;
    vecto=array[1..26] of real;
var aa,ba,ca,a,ai,bi,y2,ti,kr,xp,xs,xr,xw,xx,xsur,yd,yp,ys,yn,xd,
    rc,am,rch,pch,tkr,pkr:wektor;
    wistabe;
    rc,rg:tablica;
    tau,tf,fs:vect;
    eo,xsr,xfi:tabel;
    mwstabil;
    d1,d2:vecto;
($I as\polki\podprogr\potega.pas)
($I as\polki\podprogr\intablic.pas)
($I as\polki\podprogr\inter2.pas)
($I as\polki\podprogr\sprawno.pas)
($I as\polki\podprogr\fizchem.pas)
($I as\polki\podprogr\wybred.pas)
($I as\polki\podprogr\krytwyb.pas)
($I as\polki\podprogr\kryteria.pas)
function Fmax:real;
    var ha:real;
    begin
        if u=1 then ha:=am else ha:=ai;
        ha:=ha-gz1-g-0.006;
        fmax:=3.148ha$dz
    end; (function Fmax)
procedure zwieksz;

```

```

begin
  case n of
    1: if i=id then i:=0
      else begin
        i:=i+1;
        d:=d1(i)
      end;
    2: if i=ip then i:=0
      else begin
        i:=i+1;
        d:=d2(i)
      end;
  end; (case n)
end; (procedure zwieksz)
begin
  ClrBcr;
  assign(danzawor,'a:\polki\zbiordan\danawor.pas');
  reset(danzawor);
  readln(danzawor,pcal,mp,ih,id,ip,k,as);
  assign(databzaw,'a:\polki\zbiordan\databzaw.pas');
  reset(databzaw);
  begin
    for i:=1 to 18 do
      for j:=1 to 4 do
        for ii:=1 to 10 do
          read(databzaw,nb[i,j,ii]);
        for i:=1 to 16 do
          for j:=1 to 3 do
            for ii:=1 to 12 do
              read(databzaw,nb[i,j,ii]);
            for i:=1 to 5 do
              for j:=1 to 14 do read(databzaw,e1[i,j]);
            for i:=1 to 14 do read(databzaw,y2[i]);

```



```

for i:=1 to 3 do read(databzaw,bi(i));
for i:=1 to 10 do read(databzaw,hi(i));
for i:=1 to 18 do read(databzaw,di(i));
for i:=1 to 16 do read(databzaw,d2(i));
for i:=1 to 7 do read(databzaw,hpi(i));
for i:=1 to 11 do read(databzaw,mi(i));
for i:=1 to 11 do read(databzaw,mi1(i));
for i:=1 to 3 do read(databzaw,ga(i));
for i:=1 to 3 do read(databzaw,aa(i));

end; (czytanie)

writeln('czy obliczana bedzie polka z zaworkami V1? t/n ');
readln(typz);

if typz='t' then tz:=1 else tz:=4;

if tz=1 then typz:='zaworki typu V1'
      else typz:='zaworki typu V4';

fizchem;
kryteria;
poczliwyb:=1;

v:=vg83600;l:=vc83600;vo:=v8sqrt((rog/(roc-rog)));

if (vo<950) and (l<250) or (vo<2000) and (l<150) then
begin x:=(0.0428*vo+72.6687)/(1.9017+0.00058*vo-0.001875*l);
      y:=1.90178*x-72.6687;
      z:=sqrt(sqr(x-39)+sqr(y-12));
      dor:=0.056828*z+0.5;
      n:=1;
end else
begin x:=(0.0428*vo+174.7258)/(5.4939+0.00058*vo-0.001875*l);
      y:=5.49398*x-174.7258;
      z:=sqrt(sqr(x-35)+sqr(y-18));
      dor:=0.031828*z+1.5;
      n:=2;
end;

if n=1 then nstrum:='polka jednostrumieniowa'

```

```

else nstrum:="polka dwustrumieniowa";
assign(wynzawor,"cs\polki\wynzawor.pjs");
rewrite(wynzawor);
if (dan="t") or (bile="t") then
begin
  writeln(wynzawor,"");
  writeln(wynzawor,"
                                WYNIKI OBLICZEN POLEK',
                                ' ZAMORKOWYCH");

  writeln(wynzawor,"
                                dla zalozonych parametrow rucho',
                                ' wych (Vg,Vc)");
  writeln(wynzawor,"");
  writeln(wynzawor,"
                                Vg=' ,vg:6:4,' m3/s',
                                '
                                Vc=' ,vc:7:5,' m3/s');
  writeln(wynzawor,"");
  writeln(wynzawor,"
                                ',typz,'
                                ',nstrum);
  writeln(wynzawor,"
                                ');
  writeln(wynzawor,"");
end
else
begin
  writeln(wynzawor,"");
  writeln(wynzawor,"
                                WYNIKI OBLICZEN POLEK',
                                ' ZAMORKOWYCH");

  writeln(wynzawor,"
                                dla zalozonych parametrow rucho',
                                ' wych (Vg,Vc,R)");
  writeln(wynzawor,"");
  writeln(wynzawor,"
                                Vg=' ,vg:6:4,' m3/s',
                                '
                                Vc=' ,vc:7:5,' m3/s',
                                '
                                R=' ,r:4:1);
  writeln(wynzawor,"");
  writeln(wynzawor,"
                                ',typz,'
                                ',nstrum);
  writeln(wynzawor,"
                                ',rek);

```

```

    writeln(wybzawor,'');
end;

assign(wybzawor,'c:\polki\wybzawor.pas');
rewrite(wybzawor);

if (dan='t') or (bile='t') then
begin
    writeln(wybzawor,'');
    writeln(wybzawor,'',
        '
        WYNIKI OBLICZEN POLEK',
        'ZAWORKOWYCH');

    writeln(wybzawor,'',
        '
        dla zalozonych parametrow rucho',
        'wych (Vg,Vc)');

    writeln(wybzawor,'');
    writeln(wybzawor,'
        Vg=',vg:6:4,' m3/s',
        '
        Vc=',vc:7:5,' m3/s');

    writeln(wybzawor,'');
    writeln(wybzawor,'
        ',typz,'
        ',nstrum);
    writeln(wybzawor,'
        ');
    writeln(wybzawor,'');

end
else
begin
    writeln(wybzawor,'');
    writeln(wybzawor,'',
        '
        WYNIKI OBLICZEN POLEK',
        'ZAWORKOWYCH');

    writeln(wybzawor,'',
        '
        dla zalozonych parametrow rucho',
        'wych (Vg,Vc,R)');

    writeln(wybzawor,' ');

```

```

writeln(wybzawor,'          Vg=',vg:4:4,' m3/s',
          '          Vc=',vc:7:5,' m3/s','          R=',r:4:1);

writeln(wybzawor,'');
writeln(wybzawor,'          ',typz,'          ',nstrum);
writeln(wybzawor,'          ',rok);
writeln(wybzawor,'');
end;

lc:=0.75*dor/n;
if dor<0.9 then ff:=0.75 else ff:=0.82;
fmi:=3.148*mi*dz;
w1:=0.1698*wp;
w2:=0.006748*wp*sqrt(roc-rog);
j:=1;ik:=1;
poc1:hi=h1[j];
fv:=a[j]*rog+m1[j];
fam:=(vo+1.3618181c)/(3600*fv*wp*ff);
w3:=0.0088*wp*sqrt(h1(roc-rog));
if (w1>w2) and (w1>w3) then wc:=w1
                        else if (w2>w1) and (w2>w3) then wc:=w2
                        else wc:=w3;

spm:=vc/wc/ff;
a2:=0.118*fam;
a4:=28*spm;
if spm<a2 then else goto efka;
if a4<a2 then spm:=a4 else spm:=a2;
efm: fkm:=fam+a4;
dm:=1.12848*sqrt(fkm);
d:=dm;
if n=1 then srednica(d1,id,i) else srednica(d2,ip,i);
if i=0 then
begin
writeln(wynzawor,'***H=',h:4:2,'***',
          ' srednica wieksza od',

```

```

      * znormalizowanej');
      writeln(wynzawar, '');
      goto e)
    end;

    hp:=0.0;
ed1: f:=0.78348d8d;
    bn:=1;
    wp:=wq/f;
et:  t:=0.076;
    nt:=1;
    if n=1 then t:=5 else t:=8;
    case n of
1:begin
    ni:=4;
    sp:=nb[i,bn,1];
    lp:=nb[i,bn,2];
    fp:=nb[i,bn,3];
    fa:=nb[i,bn,4];
    b:=lp/d;
    end;
2:begin
    if d<=-4.0 then ni:=3 else ni:=2;
    ad:=nb[i,bn,1];
    sp1:=nb[i,bn,2];
    sp2:=nb[i,bn,3];
    lp1:=nb[i,bn,4];
    lp2:=nb[i,bn,5];
    fpb:=nb[i,bn,6];
    fpe:=nb[i,bn,7];
    fa:=nb[i,bn,12];
    b:=lp1/d;
    lp:=lp1+lp2;
    sp:=sp1+sp2/2;

```



```

      fp:=(fpa+fpb)/2
    end;
  end;(case n)
    si:=(d-2*ap)/2;
    bp:=(fa-fp)/2/s;
    if (spa/fam)<=(fp/fa) then goto ebp else goto eibn;
eibn:if bn=ni then else begin
ebn:
      bn:=bn+1; goto et;
    end;
  zwieksz;
  if i=0 then
    begin writeln(wynzawor,'888H-',h14:2,' D-',
      d14:2,'888 za mala',
      ' droga przeplywu cieczy');
      writeln(wynzawor,'');
      goto ej;
    end else goto ed1;
ebp: wa:=vg/fa;
enz: if n=1 then
  begin
    nz:=nb[i,bn,t1];
    fo:=nb[i,bn,t1+1];
  end else
    begin
      nz:=nb[i,bn,t1];
      fo:=nb[i,bn,t1+1];
    end;
  ni:=nz;
  ui:=1;
  az:=na;
  fz:=0.76548dz8dz8nz;
  w:=vg/fo;
  n2:=0.0;qz2:=0.0;az2:=0.0;

```

```

gx1:=ga[u];
ax1:=aa[u];
ax:=ax1;
if tz=1 then else goto ewo;
y1=fo/fz;
a2:=fa18nz/fo;
k1:=3300$pot(a2,-1.2);
k2:=6.71$pot(y,-2.49)$exp(5.91$y);
ew: x1:=faax8nz/fo;
k3:=920$pot(x,-2.95);
k4:=2160$pot(x,-0.8);
z:=sqrt(ax8nz/rog/fo);
wo1:=2.819$z$sqrt(k2/k1);
wo2:=5.094$z$sqrt(k2$h/(k4-k3));
eifw1<0.78w>wo1 then else
  if (n=1)and(nt=3)or(n=2)and(nt=2) then
    begin
      if bn=ni then
        begin writeln(wynzawor,'$$$H=',h14:2,
          ' D=',d14:2,
          ' lp/D=',b16:4, ' $$$',
          ' za mala predkosc gazu',
          ' w otworkach');
          writeln(wynzawor,'');
          goto ej
        end else goto ebn
      end else begin
        t1:=t+0.038; nt:=nt+1;
        t1:=t1+2; goto enz
      end;
    if 1.3$w<wo2 then
      eweg:begin
        ey1: y1:=813.6$vc/pot(lp,2.5);

```

```

    if ye<7.2 then
edri: begin
    writeln(wynzawor,'$$$H=',h:4:2,' D=',d:4:2,' Lp='
    ,lp:4:2,'$$$ parametr E poza zakresem',
    ' interpolacji');
    writeln(wynzawor,'');
    goto ej;
end;
if ye>30 then begin if bn=ni then goto edri else goto ebn end;
e:=indwa(ye,b,y2,b1,e1,14,5);
ehs: a2:=vc/lp;
hs:=0.667*es*pot(a2,0.667);
if 2*es>3.0 then begin hsp:=0.0375; goto eap end
    else begin if (2*es)<1.5 then begin
                                                    hsp:=0.0125;goto eap
                                                    end
                                                    else begin
eap:
                                                    hsp:=0.025;
                                                    ap:=1.5*hs+hsp;
                                                    ik:=1;goto ehp
                                                    end;
    end ;
    end else
begin
    if (u=im) and (az=1) then
        begin
            zwieksz;
            if i=0 then begin
                writeln(wynzawor,'$$$H=',h:4:2,
                ' D=',d:4:2,' lp/D=',b:6:4,
                '$$$ za duza predkosc',
                ' gazu w otworach');
                writeln(wynzawor,'');
            end;
        end;
    end;
end;

```

```

                                goto e;
                                and else goto ed1;
                                and else begin
                                    if az=1 then
                                        begin
                                            uz=u+1;
                                            az:=na;
en2:                                n2:=trunc(nz/az);
                                    n1:=nz-n2;
                                    if u=1a then q:=u-1
                                        else q:=u;
                                    az:=(aa[q]*n1+aa[q+1]*n2)/nz;
                                    az1:=aa[q];
                                    az2:=aa[q+1];
                                    qz1:=ga[q];
                                    qz2:=ga[q+1];
                                    if n1=0 then
                                        begin
                                            az1:=0.0;
                                            qz1:=0.0
                                        end;
                                if tz=1 then goto ez
                                    else
                                        begin
enw:                                w01:=0.183*sqrt((roc-rog)/rog);
                                    w02:=sqrt(6000*az/(rog*(-0.4316*faax/dot/dot+1.267)));
                                    goto eifu
                                        and
                                            and else begin az:=az-1;
                                                goto en2
                                            end;
                                        end;
                                    end;
                                end;
                                end;

```



```

a4:=vc/fa;
pc:=4.0528pot(1.2,-u)*
pot(a2,-0.25)*
pot(hp,0.85)*pot(a4,0.35);
goto epog;
end;

end;

end;

if u=uo2 then else
if u>uo1 then
begin
psi=(k28a8nz/(f08roc)+k38rog8u8w/(9.818roc))*1e-3;
goto epc;
end else
begin
psi=1e-38k18rog8u8w/(9.818roc);
goto epc1
end;
epa: psi=1e-38k48rog8u8w/9.81/roc;
epc: a2:=fo/f; a4:=vc/fa;
pc:=1.68298pot(a2,-0.25)*pot(hp,0.85)*pot(a4,0.35);
epog: pog:=pc+ps;
hpe:=pog+pr+hp+hs+0.005;
if (hpe/hp)<=(0.59h) then
else
begin
writeln(wynzawar,'ss8H=',h14:2,
'D=',d14:2,' lp/D=',b16:4,
'hpe=',hp8:3,'ss8 ');
writeln(wynzawar,'za duze',
' spiętrzenie cieczy w przewlocie');
writeln(wynzawar,'');
goto e;

```

```

end;

ta:=0.8*(h*(h+hp-hp0));
a1:=0.6*(d/2-s);
if ta>a1 then
begin
  writeln(wynzawor,'***H=',h:4:2,' D=',d:4:2,
    ' hp=',hp:4:2,' lp/D=',b:4:2,'***');
  writeln(wynzawor,'      nierowny przeplyw cieczy');
  writeln(wynzawor,'');
  goto ekon;
end;

x:=vg*sqrt(rog)/fa;
a2:=sqrt(fo/fa)*sin(roc-rog);
y:=4.424*pot(a2,0.25);
hpl:=hp+0.65*pot(hs,0.3333)*sqrt(x/(y-x));
if hpl>h then
begin
  writeln(wynzawor,'***H=',h:4:2,' D=',d:4:2,
    '*** za mala odleglosc miedzy polkami');
  writeln(wynzawor,''); goto ej;
end;

kpo:=4.0987*pot((d+1.524),3.45);
kpi:=20.83*pot((d+3.049),2.21);
kaki=kpo+kpi;
a2:=wa/(h-hpl);
me:=4.693*pot(a2,2.31)/sinle-6;
if fo>(fmax*sz) then af:=fmax*sz else af:=fo;
a2:=wa*sqrt(rog)*fa/af;
rop:=1.5686*pot(a2,-0.628);
a1:=fo/fa;
hc:=hpi*(1-pot((x/y),0.28));
zpi:=hc*fa+hp*fp;
sprawnoz;eki=1.38ek;

```

```

krs=kryt;
pog=pog$9.81$roc;pc=pc$9.81$roc;
ps=ps$9.81$roc;pr=pr$9.81$roc;
writeIn(wynzawar,"");
writeIn(wynzawar,"$$$H=",h$4:2,"a',"      D=",d$4:2,"a',"      ",
          "F=",f$6:4,"a2',"      wg=",wg$6:4,"a/a");
writeIn(wynzawar,"");
writeIn(wynzawar,"      KPO=",kpo$5:2,"      KP=",kp$6:2,
          "      KBK=",k$6:2);
writeIn(wynzawar,"      Ek=",ek$6:4,"      KRW=",
          krs7:2);
writeIn(wynzawar,"");
writeIn(wynzawar,"  $$hp=",hp$5:3,"a',"      lp=",lp$5:3,"a");
writeIn(wynzawar,"");
writeIn(wynzawar,"      fp      s      bp",
          "      fa      faf  ");
writeIn(wynzawar,"      a2      a      a",
          "      a2      a2  ");
writeIn(wynzawar,fp$11:4,s$13:3,bp$13:3,fa$12:3,faf$13:3);
writeIn(wynzawar,"      hpp      hs      hsp",
          "      hps      fo      ");
writeIn(wynzawar,"      a      a      a",
          "      a      a2  ");
writeIn(wynzawar,hpp$11:3,hs$13:3,hsp$13:3,hps$12:3,fo$13:3);
writeIn(wynzawar,"      n1      n2      gz1",
          "      gz2      az  ");
writeIn(wynzawar,"      azt      azt      o",
          "      a      k6");
writeIn(wynzawar,n1$11:0,n2$13:0,gz1$13:4,gz2$12:4,az$13:3);
writeIn(wynzawar,"      t      ns      wc",
          "      wo      fz  ");
writeIn(wynzawar,"      a      a/a",
          "      a/s      a2  ");

```

```

writeln(wynzawor, ts:11:4, ns:13, ws:13:4, w:12:3, fs:13:3);
writeln(wynzawor, '          pog          pr          pc',
          '          ps          ta');
writeln(wynzawor, '          N/m2          N/m2          N/m2',
          '          N/m2          m');
writeln(wynzawor, pog:11:3, pr:13:4, pc:13:3, ps:12:3, ts:13:4);
writeln(wynzawor, '          hpi          rop          hc',
          '          m          zp');
writeln(wynzawor, '          a          m',
          '          k6/k6          m3');
writeln(wynzawor, hpi:11:4, rop:13:4, hc:13:3, m:12:4, zp:13:4);
writeln(wynzawor, '#####',
          '#####');
writeln(wynzawor, ''); writeln(wynzawor, ''); writeln(wynzawor, '');
if (wyb=1) or (wyb=krw) then
  append(wybzawor) else if (wyb>krw) then
    begin
      assign(wybzawor, 'c:\polki\wybzawor.pas');
      rewrite(wybzawor);
      if (dan='t') or (bila='t') then
        begin
          writeln(wybzawor, '');
          writeln(wybzawor, '          WYNIKI OBLICZEN POLEK',
                    '          ZAWORKOWYCH');
          writeln(wybzawor, '          dla zalozonych parametrow',
                    '          ruchomych (Vg,Vc)');
          writeln(wybzawor, '');
          writeln(wybzawor, '          Vg=',vg:6:4,' m3/s',
                    '          Vc=',vc:7:5,' m3/s');
          writeln(wybzawor, '');
          writeln(wybzawor, '          ',typz,'          ',nstrum);
          writeln(wybzawor, '          ');
          writeln(wybzawor, '');

```

```

end
else
begin
  writeln(wybzawor,'');
  writeln(wybzawor,'
                                WYNIKI OBLICZEN POLEK',
                                'ZAWORKOWYCH');
  writeln(wybzawor,'
                                dla zalozonych parametrów',
                                'ruchomych (Vg,Vc,R)');
  writeln(wybzawor,'');
  writeln(wybzawor,'
                                Vg=',vg:6:4,' m3/s',
                                '
                                Vc=',vc:7:5,' m3/s',
                                '
                                R=',r:4:1);
  writeln(wybzawor,'');
  writeln(wybzawor,'
                                ',typz,'
                                ',nstrum);
  writeln(wybzawor,'
                                ',rek);
  writeln(wybzawor,'');
end;

end else goto swyb;

writeln(wybzawor,'');
writeln(wybzawor,'
                                $$$H=',h:4:2,' m',
                                '
                                D=',d:4:2,' m',
                                '
                                F=',f:6:4,' m2',
                                '
                                wg=',wg:6:4,' m/s');
writeln(wybzawor,'');
writeln(wybzawor,'
                                KPD=',kpo:5:2,'
                                KP=',kp:6:2,'
                                '
                                KSK=',kek:6:2);
writeln(wybzawor,'
                                Ek=',ek:6:4,'
                                KRW=',
                                kwi:7:2);
writeln(wybzawor,'');
writeln(wybzawor,'
                                $$$hp=',hp:5:3,' m',
                                '
                                lp=',lp:5:3,' m');
writeln(wybzawor,'');
writeln(wybzawor,'
                                fp
                                s
                                bp',
                                '
                                fa
                                fmf
                                ');
writeln(wybzawor,'
                                m2
                                m
                                m',
                                '
                                m2
                                m2
                                ');
writeln(wybzawor,'fp:11:4,s:13:3,bp:13:3,fa:12:3,fmf:13:3);

```



```

writeln(wybzawor, '      hpp      hs      hsp',
        '      hpe      fo      ');
writeln(wybzawor, '      a      a      a',
        '      a      a2 ');

writeln(wybzawor, hpp:11:3, hs:13:3, hsp:13:3, hpe:12:3, fo:13:4);
writeln(wybzawor, '      n1      n2      gz1',
        '      gz2      sz ');
writeln(wybzawor, '      szt      szt      a',
        '      a      k5');

writeln(wybzawor, n1:11:0, n2:13:0, gz1:13:4, gz2:12:4, sz:13:3);
writeln(wybzawor, '      t      ns      wc',
        '      wo      fz ');
writeln(wybzawor, '      a      a/s',
        '      a/s      a2 ');

writeln(wybzawor, t:11:4, ns:13, wc:13:4, wo:12:3, fz:13:3);
writeln(wybzawor, '      pog      pr      pc',
        '      ps      ta');
writeln(wybzawor, '      N/a2      N/a2      N/a2',
        '      N/a2      a');

writeln(wybzawor, pog:11:3, pr:13:4, pc:13:3, ps:12:3, ta:13:4);
writeln(wybzawor, '      hpi      rop      hc',
        '      me      xp');
writeln(wybzawor, '      a      a',
        '      k6/k6      a3');

writeln(wybzawor, hpi:11:4, rop:13:4, hc:13:4, me:12:4, xp:13:4);
writeln(wybzawor, '!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!',
        '!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!');

writeln(wybzawor, '');writeln(wybzawor, '');writeln(wybzawor, '');

wyb:if (wyb=1) or (wyb>krw) then wyb:=krw;
ekon:if ik=k then else begin ik:=ik+1;goto shp; end;
ej: if j=ih then else begin j:=j+1; goto poc1; end;

close(wynzawor);close(wybzawor);

end. ( koniec programu ZAWOR )

```

4.3. Tabulogram programu BITO

```

program BITO(dane,dane1,dane2,dane3,danesito,dantabsl,tabwar,wynsito,
             wybsito);
uses CRT,printer;

var dane,dane1,dane2,dane3,danesito,dantabsl,tabwar,wynsito,
    wybsito:text;

label pocxl,pocl,edrl,eip,eye,ehp,eifc,edo,eik,eilp,eido,ekon,ee,
    ewyb;

var fat,i,ib,jd,ix,ido,ifi,ih,ii,ik,ini,ip,il,j,jfo,jk,jl,k,klr,m,
    ms,of,nf,ns,ncek,nlsk,num:integer;
    a,alfa,alfas,as,al,a2,b,bet,bp,c,ch,chi,cha,cek,d,dd,dss,dot,
    dyc,e,ek,seg,esp,etc,etca,etg,etgs,f,fa,fg,fi,fo,fp,fr,fw,fef,
    h,hc,hp,hpe,hpi,he,hpp,hs,hsl,hse,kgy,kp,kpo,kw,ks,kak,kw,l,
    lgc,ldc,lak,li,lp,se,sc,scs,sg,sgs,sig,sxy,nsl,pc,pcal,ps,pog,
    pr,ps,psa,r,roc,roca,rog,roga,rop,s,s1,si,sis,sp,susak,su,ta,
    tg,tc,tdc,tdg,tgc,tdg,tp,ts,t0,v,vml,vc,vg,vdg,vgg,w,wa,wml,
    wo,wob,wr,wsp,wyb,x,xpri,y,ye,xp:real;
    hpl:array [1..7] of real;
    lpl:array [1..3] of real;
    dol,sg:array [1..4] of real;
    norfoi:array [1..144] of real;
    hli:array [1..11] of real;
    bila,dana,jed,xnak,dan,reak,rekt:char;
    rek:string[20];
    dak:string[16];

const wll = 14;
    wve = 5;
    wlv = 20;
    wpi = 1e-6;

type wektor=array[1..wll] of real;
    tabe=array [1..5,1..14] of real;
    tablica=array [1..wll,1..wlv] of real;

```

```

    tabel=array [1..w1l,1..wv1] of real;
    vect=array[1..wv1] of real;
    tabl1=array[1..w1l,1..w1l] of real;
    vecto=array[1..26] of real;

var aa,ba,ca,b2,c1,ch1,h2,k1,k2,kr,a2,y2,t1,xp,xs,xw,xr,xh,xsw,yp,
    ys,yd,yw,xd,ro,aa,rch,pch,tkr,pkr:wektor;

    ei:tabel;
    rc,rg:tablica;
    tsu,tf,fs:vect;
    eo,xsr,xf:tabel;
    ww:tabl1;
    di:vecto;

(01 as\polki\podprogr\potega.pas)
(01 as\polki\podprogr\intablic.pas)
(01 as\polki\podprogr\inter2.pas)
(01 as\polki\podprogr\interpol.pas)
(01 as\polki\podprogr\wybered.pas)
(01 as\polki\podprogr\sprawno.pas)
(01 as\polki\podprogr\fizchem.pas)
(01 as\polki\podprogr\krytyb.pas)
(01 as\polki\podprogr\kryteria.pas)

begin
ClrScr;

    assign(danesito,'as\polki\zbiorden\danesito.pas');
    reset(danesito);

    readln(danesito,pcal,ih,jd,k,ib,ido,as);
    assign(dantabsl,'as\polki\zbiorden\dantabsl.pas');
    reset(dantabsl);

begin
    for i:=1 to 7 do read(dantabsl,c1[i]);
    for i:=1 to 7 do read(dantabsl,h2[i]);
    for i:=1 to 5 do
        for j:=1 to 14 do read(dantabsl,e1[i,j]);

```

```

for i:=1 to 14 do read(dantabsl,y2[i]);
for i:=1 to 3 do read(dantabsl,b2[i]);
for i:=1 to 10 do read(dantabsl,k1[i]);
for i:=1 to 10 do read(dantabsl,ch1[i]);
for i:=1 to 9 do read(dantabsl,s2[i]);
for i:=1 to 9 do read(dantabsl,k2[i]);
for i:=1 to 11 do read(dantabsl,h1[i]);
for i:=1 to 12 do read(dantabsl,d1[i]);
for i:=1 to 7 do read(dantabsl,hpl[i]);
for i:=1 to 4 do read(dantabsl,dol[i]);
for i:=1 to 4 do read(dantabsl,sg[i]);
for i:=1 to 3 do read(dantabsl,lpl[i]);
for i:=1 to 144 do read(dantabsl,norfo[i]);

end; (czytanie)

fizchem;
kryteria;

pocz:=assign(wybsito,'cs\polki\wybsito.pas');
rewrite(wybsito);

if (dan='t') or (bil='t') then
begin
  writeln(wybsito,'');
  writeln(wybsito,'          WYNIKI OBLICZEN POLEK',
    ' BITOWYCH');
  writeln(wybsito,'          dla zalozonych parametrow ',
    ' ruchomych (Vg,Vc)');
  writeln(wybsito,'');
  writeln(wybsito,'          Vg=',vg:4:4,' m3/s',
    '          Vc=',vc:7:3,' m3/s');
  writeln(wybsito,'          ');
  writeln(wybsito,'');
end
else
begin

```



```

writeln(wybsito,'');
writeln(wybsito,'          WYNIKI OBLICZEN POLEK',
        ' BITOWYCH');
writeln(wybsito,'          dla zalozonych parametrow ',
        ' ruchomych (Vg,Vc,R)');
writeln(wybsito,'');
writeln(wybsito,'          Vg=',vg:6:4,' m3/s', '          Vc=',
        'vc:7:5,' m3/s', '          R=',r:4:1);
writeln(wybsito,'          ',rek);
writeln(wybsito,'');
end;

assign(wynsito,'c:\polki\wynsito.pas');
rewrite(wynsito);

if (dan='t') or (bila='t') then
begin
  writeln(wynsito,'');
  writeln(wynsito,'          WYNIKI OBLICZEN POLEK',
        ' BITOWYCH');
  writeln(wynsito,'          dla zalozonych parametrow ',
        ' ruchomych (Vg,Vc)');
  writeln(wynsito,'');
  writeln(wynsito,'          Vg=',vg:6:4,' m3/s',
        '          Vc=',vc:7:5,' m3/s');
  writeln(wynsito,'          ');
  writeln(wynsito,'');
end
else
begin
  writeln(wynsito,'');
  writeln(wynsito,'          WYNIKI OBLICZEN POLEK',
        ' BITOWYCH');
  writeln(wynsito,'          dla zalozonych parametrow ',
        ' ruchomych (Vg,Vc,R)');

```



```

writeln(wynsio,'');
writeln(wynsio,'          Vg=',vg:6:4,' a3/e', '          Vc=',
          vc:7:5,' a3/e', '          R=',r:4:1);
writeln(wynsio,'          ',rek);
writeln(wynsio,'');
end;

wyb:=1;
j:=0;
dot:=0;
poci:j:=j+1;
h:=h1[j];
c:=inter(h,c1,h2,7);
w:=0.47e-54csqrt((roc-r0g)/rog);
d:=1.128sqrt(vg/w);
ik:=1;i:=1;ip:=0;a:=1;
srednica(d1,jd,jk);
if jk=0 then begin
    writeln(wynsio,'***H=',h:4:2,'srednica wieksza',
            'od znormalizowanej');
    goto skon
end;

w:=1.273vg/d/dj;
f:=0.7854sd8d;
eip: ip:=ip+1;
bi:=ip1[ip];
lp:=b8d;
li:=0.78vc;
eye: ye:=01381/pot(lp,2.5);
if ye>30 then begin if ip=ib then goto edr1 else goto eip end;
if ye>0.2 then goto ee;
edr1: begin
    writeln(wynsio,'***H=',h:4:2,' D=',d:4:2,' Lp=',lp:4:2,
            '*** parametr w poza zakresen interpolacji');

```

```

        goto ekon;
    end;
ee:  e:=indwa(ya,b,y2,b2,e1,14,5);
    a2:=1/lp;
    hs:=0.667888pot(a2,0.667);
    if abs(1-(0.78vc))<epsi then
        begin
            hsi:=hs;l:=1.38vc; goto eye
        end
    else
        if abs(1-(1.38vc))<epsi then
            begin
                hsm:=hs;
                l:=vc;
                goto eye
            end;
        if hs<0.012 then if ip=ib then
            begin
                writeln(wynsito,'ssH=',h:4:2,
                    ' D=',d:4:2,
                    ' Lp=',lp:4:2,
                    ' hs=',hs:6:4,
                    ' hp=',hp:4:2);
                writeln(wynsito,'sss za male',
                    ' spietrzanie',
                    ' na przelewie');
                goto ekon
            end else goto eip;
        ehp:=hp1[ik];
        ch:=hsi+hp;
        chi:=ch;
        cha:=hsm+hp;
        eifc:if ch<0.05 then begin

```

```

        writeln(wynsito, 'ssstH=', h14:2, ' D=', d14:2,
                ' hp=', hp:5:3, 'ssst za mala',
                ' wysokość przelewu');

        goto eik
    end
else if ch>0.1 then
    begin
        writeln(wynsito, 'ssstH=', h14:2,
                ' D=', d14:2,
                ' hp=', hp:5:3,
                ' ssst za duza',
                ' wysokość',
                ' przelewu');

        goto eik
    end;

if ch<>cha then begin ch:=cha; goto eifc end;
ch:=hs+hp;
hpp:=hp-0.006;
fr:=hpp81p;
a1:=sqrt(1-b8b);
fp:=d8d8(arctan(b/a1)-b8a1)/2;
if (0.58fp)<fr then wr:=28vc/fp else wr:=vc/fr;
pr:=0.16595*wr8wr;
if pr >=0.025 then if ip=ib then
    begin
        writeln(wynsito, 'ssstH=', h14:2,
                ' D=', d14:2,
                ' hp=', hp:5:3,
                'ssst za duza ',
                'strata ciśni',
                'enia przy ',
                ' wpływie cie',
                'czy na polke');
    end;

```

```

                                goto eik
                                and else goto eip;

a:=0.5*sqrt(dsd-lp*lp);
fa:=f-fp;
edo: dot:=dol[m];
if abs(dot-0.003)<epsi then fi:=2.667 else fi:=2.5;
jfo:=128*(jk-1)+38*(m-1)+ip;
fo:=norfo[jfo];
fw:=fo*fi*fi/0.9065;
if cha>0.1 then cha:=0.1;
kw:=inter(cha,k1,chi,10);
wai:=(kw-22.98*(1-39.48*dot))/sqrt(rog);
wo:=vg/fo;
if wai>wo then if ip<ib then goto eip
                                else begin
                                    writeln(wynsito,'$$$H=',h:4:2,
                                                ' D=',d:4:2,
                                                ' hp=',hp:5:3,
                                                ' do=',dot:6:4);
                                    writeln(wynsito,'$$$ za sala',
                                                ' predkosc gazu',
                                                ' w otworkach');
                                    goto eido
                                end;

fg:=f-0.58fp;
s1:=sg[m]/dot;
ks:=inter(s1,k2,s2,9);
wob:=28.98*sqrt(rog/roc);
bet:=-58ch-0.18wob+1.05;
pc:=bet*(cha+12.7/roc);
ps:=0.08628ks*wo*wo*rog*(0.48*(1.25-fa/fg)+sg*(1-fa/fg))/roc;
psa:=0.298ps;
pas:=0.005+0.058cha;

```

```
if pos<pa then if ip<ib then goto eip
```

```
else begin
```

```
    writeln(wynsito,'SSSH=',h:4:2,
```

```
        ' D=',d:4:2,
```

```
        ' hp=',hp:5:3,
```

```
        ' lp=',lp:4:2,
```

```
        'SSS polka nie',
```

```
        ' stabilna');

```

```
    goto eido
```

```
end;
```

```
pog:=ps+pc;
```

```
hps:=cha+pog+pr;
```

```
if (28hps-hp)>h then begin
```

```
    writeln(wynsito,'SSSH=',h:4:2,' D=',
```

```
        d:4:2,' hp=',hp:4:2,
```

```
        'SSS za mala odleg',
```

```
        'losc miedzy polkami');

```

```
    goto eik
```

```
end;
```

```
ta:=0.88(haes(h+hp-hps));
```

```
a1:=0.68(d/2-s);
```

```
if ta>a1 then begin writeln(wynsito,'SSSH=',h:4:2,' D=',d:4:2,
```

```
        ' hp=',hp:4:2,
```

```
        'SSS niezupełne umuniecie',
```

```
        ' gazu z cieczy');

```

```
    goto eik
```

```
end;
```

```
bp:=(f-fp)/2/s;
```

```
wsp:=sqrt(rqg)/fg;
```

```
hpi:=0.1146+0.758hp+9.6128vc/bp-0.0688wsp+0.0382twsp8wsp;
```

```
hc:=9.0247+0.228hp-0.011656wsp+4.5368vc/bp;
```

```
rapi=hc/hpi;
```

```
if hpi>h then begin writeln(wynsito,'SSSH=',h:4:2,' D=',d:4:2,
```


* hp=",hp:4:2,"ss za duza",

* wysokość piany");

gate aik

end;

zp:=hp18rop8fa+hp8fp/2;

kpo:=3.58648pot((d+1.524),3.45);

kp:=20.8388spot((d+3.049),2.21);

ksk:=kpo+kp;

he:=h-hp;

a2:=vg/fg/he;

ae:=5.7e-6spot(a2,3.2)/s1;

sprawosc;

krw:=kryt;

v:=9.818roc;

pc:=pcsv;ps:=psv;psa:=psav;psa:=psv;pog:=pogsv;pr:=prsv;

fp:=fp/2;

writeln(wynsito,"");

writeln(wynsito,"ssH=",h:4:2,"m", " D=",d:4:2,"m",

" F=",f:6:4,"m2", " wg=",w:6:4,"m/s");

writeln(wynsito,"");

writeln(wynsito," KPO=",kpo:5:2," KP=",kp:6:2,

" KSK=",ksk:6:2);

writeln(wynsito," Ek=",ek:6:4," KRW=",

krw:7:2);

writeln(wynsito,"");

writeln(wynsito," ssHp=",hp:5:3,"m", " lp=",lp:5:3,"m");

writeln(wynsito,"");

writeln(wynsito," fp s bp",

" fa fg");

writeln(wynsito," m2 m m",

" m2 m2");

writeln(wynsito,fp:12:4,s:13:3,bp:13:3,fa:12:3,fg:13:3);

writeln(wynsito," hpp hs hsi",


```

begin
  writeln(wybsito,'');
  writeln(wybsito,'          WYNIKI OBLICZEN POLEK',
            '          SITOWYCH');
  writeln(wybsito,'          dla zalozonych parametrow ',
            '          ruchowych (Vg,Vc)');
  writeln(wybsito,'');
  writeln(wybsito,'          Vg=',vg:6:4,' m3/s',
            '          Vc=',vc:7:5,' m3/s');
  writeln(wybsito,'          ');
  writeln(wybsito,'');
end
else
begin
  writeln(wybsito,'');
  writeln(wybsito,'          WYNIKI OBLICZEN POLEK',
            '          SITOWYCH');
  writeln(wybsito,'          dla zalozonych parametrow ',
            '          ruchowych (Vg,Vc,R)');
  writeln(wybsito,'');
  writeln(wybsito,'          Vg=',vg:6:4,' m3/s',          Vc=',
            '          vc:7:5,' m3/s',          R=',r:4:1);
  writeln(wybsito,'          ',rek);
  writeln(wybsito,'');
end;
and else goto swyb;
writeln(wybsito,'');
writeln(wybsito,'SSSH=',h:4:2,'m',          D=',d:4:2,'m',
            '          F=',f:6:4,'m2',          ug=',u:6:4,'m/s');
writeln(wybsito,'');
writeln(wybsito,'          KPO=',kpo:5:2,'          KP=',kp:6:2,
            '          KSK=',kks:6:2);
writeln(wybsito,'          'Ek=',ek:6:4,'          KPM=',

```

```

krs:7:2);

writeln(wybsito,"");

writeln(wybsito," 88hp=",hp:5:3,"a',"      1p=",1p:5:3,"a");

writeln(wybsito,"");

writeln(wybsito,"      fp      a      bp",
"      fa      fg");

writeln(wybsito,"      a2      a      a",
"      a2      a2");

writeln(wybsito,fp:12:4,a:13:3,bp:13:3,fa:12:3,fg:13:3);

writeln(wybsito,"      hpp      ha      hai",
"      hsa      cha");

writeln(wybsito,"      a      a      a",
"      a      a");

writeln(wybsito,hpp:12:3,ha:13:3,hai:13:3,hsa:12:3,cha:13:3);

writeln(wybsito,"      do      fi      fo",
"      fw      na");

writeln(wybsito,"      a      a2",
"      a2");

writeln(wybsito,dot:12:4,fi:13:3,fo:13:3,fw:12:3,na:13);

writeln(wybsito,"      wai      wo      wr",
"      hpe      pr");

writeln(wybsito,"      a/s      a/s      a/s",
"      a      N/a2");

writeln(wybsito,wai:12:2,wo:13:2,wr:13:2,hpe:12:2,pr:13:3);

writeln(wybsito,"      ps      psa      pa",
"      pc      pog");

writeln(wybsito,"      N/a2      N/a2      N/a2",
"      N/a2      N/a2");

writeln(wybsito,ps:12:3,psa:13:3,pa:13:3,pc:12:3,pog:13:3);

writeln(wybsito,"      ta      hpi      hc",
"      aa      xp");

writeln(wybsito,"      a      a      a",
"      kb/kb      a3");

```

```

writeln(wybsito, tas:12:3, hpi:13:3, hc:13:3, mm:12:3, xp:13:4);
writeln(wybsito, '#####',
              '#####');
writeln(wybsito, ''); writeln(wybsito, '');
. writeln(wybsito, '');
close(wybsito);

swyb: if (wyb=1) or (wyb>krw) then wyb:=krw;
wlip: if ip<ib then goto wip;
eido: if a<ido then begin a:=a+1; goto edo end;
eik: if ik<k then begin ik:=ik+1; goto ehp end;
ekan: if j<ih then goto paci;

close(wynsito); close(wybsito);

end. ( koniec programu B1T0 )

```


4.4. Tabulogram programu SITBEZ

```

program SITBEZ(dane,dane1,dane2,dane3,dasitbez,tabsibez,tabwar,
               wunsibez, wysibez);
uses CRT,printer;

var dane,dane1,dane2,dane3,dasitbez,tabsibez,tabwar,wunsibez,
    wysibez:text;

label pocz1,poc1,poci,efi,efik,eh,eik,ewyb;

var i,ii,ib,jd,id,ifi,ido,ih,ik,ini,ip,inb,iw,i5,j,jfo,jk,jkl,k,
    kkr,as,nf,ncsk,nlsk:integer;

    a,alfa,alfas,al1,al,a11,a2,a22,b,bet,c,cek,cl,d,dd,dx,dot,
    des,ek,ekl,etc,etcs,etg,etgs,f,fi,fi2,fo,h,hr,he,hpi,kp,kpo,
    krw,kai,kak,l,lgc,lck,lk,l1,ae,ac,ace,ag,ags,axy,pc,pcal,pog,
    ps,r,roc,rocs,rogs,rog,rop,s,si,sis,sa,sp,suak,su,tau,tg,tc,
    tdc,tdg,tgc,tgg,tp,ts,t0,tot,v,vc,vdg,vgg,vg,vcol,w,wa,wi,wo,
    wyb,xpri,y,y2,zp:real;

bi:array [1..7] of real;

dol,a1,fi:array [1..4] of real;

norfoi:array [1..48] of real;

hi:array [1..11] of real;

bila,dan,dana,jed,reak,rekt,sit,znak:char;

rek:string[20];

dak:string[16];

const wil = 14;

    wve = 5;

    wiw = 20;

    epsi=1e-6;

type wektor=array[1..wil] of real;

    tablica=array [1..wil,1..wiw] of real;

    tabel=array[1..wil,1..wve] of real;

    vect=array[1..wve] of real;

    tabl=array[1..wil,1..wil] of real;

    vecte=array[1..26] of real;

```

```

var aa,ba,ca,ea1,y1,ti,xp,xs,xw,xx,xsur,yd,yp,ys,yw,ro,es,rch,pch,
    tkr,pkr,kr,xr,xdisvektor;
rc,rg:tablica;
eo,xsr,xftabel;
tsu,tf,fs:vekt;
wt:tabli;
dl:vektor;

(01 a:\polki\podprogr\potega.pas)
(01 a:\polki\podprogr\interpol.pas)
(01 a:\polki\podprogr\intablic.pas)
(01 a:\polki\podprogr\wybsred.pas)
(01 a:\polki\podprogr\fizchem.pas)
(01 a:\polki\podprogr\krytyb.pas)
(01 a:\polki\podprogr\kryteria.pas)

procedure predkosc;
begin
    a1:=a11$pot(f1,0.89);
    w1:=a11$pot(y,0.445);
    w1:=a11$pot(y2,0.445)*0.0166;
end; (procedure predkosc)

procedure sprawbez;
var a,nog,eg,des1,eag,emp:real;
begin
    a:=vg$rog/vc/roc;
    nog:=12$(hpi-hr)/w;
    eg:=1-exp(-nog);
    des:=ax$ta$mc/eg;
    eag:=(exp(des$eg)-1)/des;
    des1:=ac$ta;
    emp:=eag/(1+eag$des1);
    ek:=ln(1+emp$(des-1))/ln(des);
end;

begin

```

ClrScr;

assign(dasitbez,'a:\polki\zbiordan\dasitbez.pas');

reset(dasitbez);

readln(dasitbez,pcal,ih,jd,ib,ido,ifi);

assign(tabsibez,'a:\polki\zbiordan\tabsibez.pas');

reset(tabsibez);

begin

for i:=1 to 9 do read(tabsibez,yi[i]);

for i:=1 to 9 do read(tabsibez,self[i]);

for i:=1 to 11 do read(tabsibez,h1[i]);

for i:=1 to 12 do read(tabsibez,d1[i]);

for i:=1 to 4 do read(tabsibez,dol[i]);

for i:=1 to 4 do read(tabsibez,sl[i]);

for i:=1 to 4 do read(tabsibez,fi1[i]);

for i:=1 to 48 do read(tabsibez,norfo[i]);

end; (czytanie)

fizchem;

kryteria;

pocz1:assign(wybsibez,'c:\polki\wybsibez.pas');

rewrite(wybsibez);

if (dan='t') or (bila='t') then

begin

writeln(wybsibez,'');

writeln(wybsibez,' WYNIKI OBLICZEN POLEK',
' BITOWYCH BEZ PRZELEWU');writeln(wybsibez,' dla zalozonych parametrow',
' ruchomych (Vg,Vc)');

writeln(wybsibez,'');

writeln(wybsibez,' Vg=',vg:6:4,' m3/s',
' Vc=',vc:7:5,' m3/s');

writeln(wybsibez,' ');

writeln(wybsibez,'');

writeln(wybsibez,'');

```

end
else
begin
  writeln(wybsibez,'');
  writeln(wybsibez,'          WYNIKI OBLICZEN POLEK',
            ' BITOWYCH BEZ PRZELEWU');
  writeln(wybsibez,'          dla zalozonych parametrow',
            ' ruchowych (Vg,Vc,R)');
  writeln(wybsibez,'');
  writeln(wybsibez,'          Vg=',vg:6:4,' m3/s',
            '          Vc=',vc:7:5,' m3/s', '          R=',r:4:1);
  writeln(wybsibez,'          ',rek);
  writeln(wybsibez,'');
  writeln(wybsibez,'');
end;

assign(wynsibez,'c:\polki\wynsibez.pas');
rewrite(wynsibez);
if (dan='t') or (bila='t') then
begin
  writeln(wynsibez,'');
  writeln(wynsibez,'          WYNIKI OBLICZEN POLEK',
            ' BITOWYCH BEZ PRZELEWU');
  writeln(wynsibez,'          dla zalozonych parametrow',
            ' ruchowych (Vg,Vc)');
  writeln(wynsibez,'');
  writeln(wynsibez,'          Vg=',vg:6:4,' m3/s',
            '          Vc=',vc:7:5,' m3/s');
  writeln(wynsibez,'          ');
  writeln(wynsibez,'');
  writeln(wynsibez,'');
end
else
begin

```

```

writeln(wynsibez,'');
writeln(wynsibez,'          WYNIKI OBLICZEN POLEK',
        ' BITOWYCH BEZ PRZELEWU');
writeln(wynsibez,'          dla zalozonych parametrow',
        ' ruchowych (Vg,Vc,R)');
writeln(wynsibez,'');
writeln(wynsibez,'          Vg=',vg:6:4,' m3/s',
        '          Vc=',vc:7:5,' m3/s', '          R=',r:4:1);
writeln(wynsibez,'          ',rek);
writeln(wynsibez,'');
writeln(wynsibez,'');
end;

poc1:wyb1=1;
a1:=rog/roc;
bet1:=1.11$pot(a1,0.118);
sa:=vc$roc/rog/vg;
a22:=sa+0.03;
y2:=inter(sa,y1,sa1,9);
a1:=sa$ag/ec;
c1:=0.53$pot(a1,5.2);
i1:=1; i5:=0; jk1:=0;

poc1:dz:=dof1[i];
j1:=1;
if abs(dz-0.003)<epsi then c1:=0.0094 else c1:=0.0065;
y1:=c1$pot(a22,-0.78);
a1:=vg/dz;
a111:=1.703$pot(a,0.1112)$pot(roc,0.472)$pot(si,0.267)
        /pot(rog,0.605)/pot(etg,0.0712)/pot(etg,0.0623);
wfi: f1:=f11[j];
predkosc;
d1:=1.128$qrt(vg/0.95/mm);
srednica(d1,jd,jk);
if jk=0 then

```



```

begin
    writeln(wynsibez,'dz=',dz:5:3,'    fi=',fi:5:3,
            '### srednica wieksza',
            ' od znormalizowanej');

    goto efik
end;

w:=1.2738vg/d/d;
inb:=(jk-1)*4+1;
f:=0.7854tdsd;
fo:=norfo[inb];
fi:=fo/f;

if (abs(fi-fi2)<epsi) and (i=i5) and (jk=jk1) then goto efik;
i5:=i,jk1:=jk,fi2:=fi;
predkosc;

if w>(0.95sw) then
begin
    writeln(wynsibez,'dz=',dz:5:3,'    fi=',fi:5:3,
            '### predkosc wieksza od',
            ' dozwolonej');

    goto efik
end;

if w<=w1 then
begin
    writeln(wynsibez,'dz=',dz:5:3,'    fi=',fi:5:3,
            '### silny przeciek');

    goto efik
end;

if w<(0.78sw) then
begin
    writeln(wynsibez,'dz=',dz:5:3,'    fi=',fi:5:3,
            '### predkosc mniejsza',
            ' od zalecanej');

    goto efik
end;

```

```

end;

a2:=dz/s1[1];
ks1:=1.758*sqr(1-f1)*pot(a2,0.2);
a2:=sqr(sa/0.62)*rog/roc/ks1;
c:=pot(a2,0.333);
tau:=c/(1+c);
a2:=1-tau;
a1:=1-bet;
pog:=sqr(w/f1)*ks1*rog*(1-tau*a1)/2/a1/pot(a2,3.0)+28*s1/dz/a1;
hr:=pog/9.31/roc;
if hr<0.024 then
begin
    writeln(wynsibez,'dz=',dz:5:3,'    fi=',fi:5:3,
            '*** wysokosc zredukowana',
            ' mniejsza od zalecanego',
            ' minimum');
    goto efik
end;

pc1:=bet*(pog-ks1*rog*tau/2/pot(a2,3));
zp1:=hr*fi;
ws1:=vg/fo;
a2:=wtg/s1;
a1:=rog/(roc-rog);
he1:=23.8*ws*pot(a2,0.43)*pot(a1,0.285);
rop1:=0.24*pot(sa,0.14);
hpi1:=(pog-sqr(w/f1)*ks1*rog/2/sqr(1-tau))/9.81/rop/roc;
h1:=hpi+he1;
k1:=1;
eh1: if h<h1[k] then h1:=h1[k]
      else begin
          if k<ih then
              begin
                  k1:=k+1;goto eh1
              end
          end
      end

```

```

end
else
begin
    writeln(wynsibez, 'dz=' , dz:5:3,
            ' f1=' , f1:5:3, 'ss',
            ' odstep polak wiekszy',
            ' od stosowanego');
    goto efik
end;

end;

a2:=w/h0;
ae:=0.4e-58pot(a2,2.56);
ps:=pog-pc;
a2:=d+3.049;
kp:=20.8388pot(a2,2.21);
a2:=d+1.524;
kpo:=2.59188pot(a2,3.45);
dot:=dz;
if abs(dot-0.0003)<epsi then tot:=0.008 else tot:=2.58dot;
kak:=kp+kpo;
sprambez;
krw:=kryt;
writeln(wynsibez, '');
writeln(wynsibez, 'ssH=' , h:4:2, 'm', ' D=' , d:4:2, 'm',
            ' F=' , f:4:4, 'm',
            ' ug=' , w:4:4, 'm/s');
writeln(wynsibez, '');
writeln(wynsibez, ' KPD=' , kpo:5:2, ' KP=' ,
            kp:4:2, ' KOK=' , kak:4:2);
writeln(wynsibez, ' Ek=' , ek:4:4,
            ' KRM=' , krw:7:2);
writeln(wynsibez, '');
writeln(wynsibez, ' ssdo=' , dot:5:3, 'm', ' f1=' , f1:4:2);

```

```

writeln(wybsibez,'');
writeln(wybsibez,'          F          Fo          to',
        '          wain          wmax');
writeln(wybsibez,'          a2          a2          a',
        '          a/s          a/s');
writeln(wybsibez,f:11:4,fo:13:4,tot:13:3,wi:13:3,wa:14:3);
writeln(wybsibez,'          wo          oo          zp',
        '          ho          hr');
writeln(wybsibez,'          m/s          k6/k6          a3',
        '          m          a');
writeln(wybsibez,wo:11:4,oo:13:4,zp:13:3,ho:13:3,hr:14:3);
writeln(wybsibez,'          pc          ps          pog',
        '          hpi          rop');
writeln(wybsibez,'          N/a2          N/a2          N/a2',
        '          a');
writeln(wybsibez,pc:11:4,ps:13:4,pog:13:3,hpi:13:3,rop:14:3);
writeln(wybsibez,'');
writeln(wybsibez,'!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!',
        '!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!');
writeln(wybsibez,'');writeln(wybsibez,'');
if (wyb=1) or (wyb=krw) then
  append(wybsibez) else if (wyb>krw) then
    begin
      assign(wybsibez,'c:\polki\wybsibez.pas');
      rewrite(wybsibez);
    end
if (dan='t') or (bila='t') then
  begin
    writeln(wybsibez,'');
    writeln(wybsibez,'          WYNIKI OBLICZEN POLEK',
        '          BITOWYCH BEZ PRZELEW');
    writeln(wybsibez,'          dla zalozonych parametrow',
        '          ruchowych (Vg,Vc)');
    writeln(wybsibez,'');
  end

```

```

writeln(wybsibez,'                               Vg=',vg:6:4,' m3/s',
        '                               Vc=',vc:7:5,' m3/s');
writeln(wybsibez,'                               ');
writeln(wybsibez,'');
writeln(wybsibez,'');
end
else
begin
  writeln(wybsibez,'');
  writeln(wybsibez,'                               WYNIKI OBLICZEN POLEK',
        '          BITOWYCH BEZ PRZELEWU');
  writeln(wybsibez,'                               dla zalozonych parametrow',
        '          ruchomych (Vg,Vc,R)');
  writeln(wybsibez,'');
  writeln(wybsibez,'                               Vg=',vg:6:4,' m3/s',
        '                               Vc=',vc:7:5,' m3/s',
        '                               R=',r:4:1);
  writeln(wybsibez,'                               ',rak);
  writeln(wybsibez,'');
  writeln(wybsibez,'');
end;
  and else goto ewyb;
writeln(wybsibez,'');
writeln(wybsibez,' 88H=',h:4:2,' m', '          D=',d:4:2,' m',
        '          F=',f:6:4,' m',
        '          wg=',w:6:4,' m/s');
writeln(wybsibez,'');
writeln(wybsibez,'          KPO=',kpo:5:2,'          KP=',
        '          kps:6:2,'          KKK=',kkk:6:2);
writeln(wybsibez,'          Ek=',ek:6:4,
        '          KRW=',krw:7:2);
writeln(wybsibez,'');
writeln(wybsibez,' 88do=',dot:5:3,' m', '          fi=',fi:4:2);

```



```

writeln(wybsibez, '');
writeln(wybsibez, '      F      Fg      to',
        '      min      max');
writeln(wybsibez, '      a2      a2      a',
        '      a/s      a/s');
writeln(wybsibez, f:11:4, fc:13:4, tot:13:3, w:13:3, wu:14:3);
writeln(wybsibez, '      wo      ao      zp',
        '      ho      hr');
writeln(wybsibez, '      a/s .      k6/k6      a3',
        '      a      a');
writeln(wybsibez, w:11:4, a:13:4, zp:13:3, h:13:3, hr:14:3);
writeln(wybsibez, '      pc      ps      pog',
        '      hpl      rop');
writeln(wybsibez, '      N/a2      N/a2      N/a2',
        '      a');
writeln(wybsibez, pc:11:4, ps:13:4, pog:13:3, hpl:13:3, rop:14:3);
writeln(wybsibez, '');
writeln(wybsibez, '#####',
        '#####');
writeln(wybsibez, '');
writeln(wybsibez, '');
wyb: if (wyb=1) or (wyb>krw) then wyb:=krw;
efi: j:=j+1;
      if j<=ifi then goto efi;
efk: i:=i+1;
      if i<=ib then goto efi;
close(wynsibez); close(wybsibez);
end. {koniec programu BITBEZ}

```

6.3. Tabulogram programu RUBIT

```

program RUBIT(dane,dane1,dane2,dane3,daruszt,tabruszt,tabwar,
              wynruszt,wybruszt);
uses CRT,printer;
var dane,dane1,dane2,dane3,daruszt,tabruszt,tabwar,wynruszt,
    wybruszt:text;
label pocz1,pocl,poci,efi,efik,eh,eik,ewyb;
var i,ii,ib,jd,id,ifi,ido,ih,ik,ini,ip,inb,iw,j,jfo,jk,k,kkr,ms,
    nf,nck,nlsk:integer;
    a,alfa,ai1,ai,aii1,a2,a22,b,bet,c,cek,ci,d,des,dd,dz,dot,ek,
    etc,etcs,etg,etgs,f,fi,fo,h,hr,he,hpi,kp,kpo,krw,ksi,kak,l,
    lgc,ldc,lsk,li,me,mc,mcs,mg,mgx,my,pc,pcal,pog,ps,r,roc,rocs,
    rogs,rog,rop,si,sis,sa,sp,suak,su,tau,s,tg,tc,tdc,tdg,tgc,tgg,
    tp,ts,t0,tot,v,vc,vg,vmol,vdg,vgg,w,wm,wi,wo,wyb,xpri,y,y2,xp,
    alfas,eki:real;
    bi:array [1..7] of real;
    dol,sl,fi:array [1..4] of real;
    hi:array [1..11] of real;
    bila,dan,dana,jed,reak,rekt,sit,xnak:char;
    rek:string[20];
    dak:string[16];
const wil = 14;
    wve = 5;
    wiw = 20;
    eps1 = 1e-6;
type wektor=array[1..wil] of real;
    tablica=array [1..wil,1..wiw] of real;
    tabel=array[1..wil,1..wve] of real;
    vect=array[1..wve] of real;
    tabl=array[1..wil,1..wil] of real;
    vecto=array[1..26] of real;
var aa,ba,ca,sai,yi,ti,xp,xs,xw,xx,xsur,yp,ys,ro,mw,rch,pch,tkr,

```

```
pk,kr,xr,xd,yd,yw:vektor;
```

```
rc,rg:tablica;
```

```
eo,xer,xf:tabel;
```

```
tsu,tf,fs:vekt;
```

```
mw:tabl1;
```

```
dis:vekt;
```

```
(@I a:\polki\podprogr\potega.pas)
```

```
(@I a:\polki\podprogr\interpol.pas)
```

```
(@I a:\polki\podprogr\intablic.pas)
```

```
(@I a:\polki\podprogr\wybrsed.pas)
```

```
(@I a:\polki\podprogr\fizchem.pas)
```

```
(@I a:\polki\podprogr\krytyb.pas)
```

```
(@I a:\polki\podprogr\kryteria.pas)
```

```
procedure predkos;
```

```
begin
```

```
  a1:=a11spot(f1,0.89);
```

```
  wa:=a11spot(y,0.445);
```

```
  w1:=a11spot(y2,0.445)*0.0166;
```

```
end; (procedure predkos)
```

```
Procedure sprambez;
```

```
var a,nog,eg,des1,esg,esp:real;
```

```
begin
```

```
  a:=vg8rog/vc/roc;
```

```
  nog:=128*(hpl-tr)/w;
```

```
  eg:=1-exp(-nog);
```

```
  des1:=mystatoc/eg;
```

```
  esg:=(exp(des1*eg)-1)/des1;
```

```
  des1:=esg;
```

```
  esp:=esg/(1+esg*des1);
```

```
  ek1:=-ln(1+exp(des1-1))/ln(des1);
```

```
end;
```

```
begin
```

```
  clrscr;
```

```

assign(daruszt,'c:\polki\zbior\den\daruszt.pas');
reset(daruszt);
readln(daruszt,pcal,ih,jd,ib,ida,ifi);
assign(tabruszt,'c:\polki\zbior\den\tabruszt.pas');
reset(tabruszt);
begin
    for i:=1 to 9 do read(tabruszt,yi[i]);
    for i:=1 to 9 do read(tabruszt,ael[i]);
    for i:=1 to 11 do read(tabruszt,h1[i]);
    for i:=1 to 12 do read(tabruszt,d1[i]);
    for i:=1 to 4 do read(tabruszt,dol[i]);
    for i:=1 to 4 do read(tabruszt,s1[i]);
    for i:=1 to 4 do read(tabruszt,f11[i]);
end; (czytanie)

fizchee;
kryteria;
pocz1:=assign(wybruszt,'c:\polki\wybruszt.pas');
rewrite(wybruszt);
if (dan='t') or (bila='t') then
begin
    writeln(wybruszt,'');
    writeln(wybruszt,'
                                WYNIKI OBLICZEN POLEK',
                                ' RUSZTOWYCH');
    writeln(wybruszt,'
                                dla zalozonych parametrow',
                                ' ruchomych (Vg,Vc)');
    writeln(wybruszt,'');
    writeln(wybruszt,'
                                Vg=',vg:6:4,' m3/s',
                                '
                                Vc=',vc:7:5,' m3/s');
    writeln(wybruszt,'
                                ');
    writeln(wybruszt,'');
    writeln(wybruszt,'');
end
else

```

```

begin
  writeln(wybruszt,'');
  writeln(wybruszt,'
WYNIKI OBLICZEN POLEK',
'
RUBIZTOWYCH');
  writeln(wybruszt,'
dla zalozonych parametrow',
'
ruchowych (Vg,Vc,R)');
  writeln(wybruszt,'');
  writeln(wybruszt,'
Vg=',vg:6:4,' m3/s',
'
Vc=',vc:7:5,' m3/s',
'
R=',r:4:1);
  writeln(wybruszt,'
',rek);
  writeln(wybruszt,'');
  writeln(wybruszt,'');
end;

assign(wynruszt,'c:\polki\wynruszt.pas');
rewrite(wynruszt);

if (dan='t') or (bila='t') then
begin
  writeln(wynruszt,'');
  writeln(wynruszt,'
WYNIKI OBLICZEN POLEK',
'
RUBIZTOWYCH');
  writeln(wynruszt,'
dla zalozonych parametrow',
'
ruchowych (Vg,Vc)');
  writeln(wynruszt,'');
  writeln(wynruszt,'
Vg=',vg:6:4,' m3/s',
'
Vc=',vc:7:5,' m3/s');
  writeln(wynruszt,'
');
  writeln(wynruszt,'');
  writeln(wynruszt,'');
end
else
begin
  writeln(wynruszt,'');
  writeln(wynruszt,'
WYNIKI OBLICZEN POLEK',

```



```

      * RUSZCZYCH');
      writeln(wynruszt,'          dla zalezonych parametrow',
      * ruchowych (Vg,Vc,R)');
      writeln(wynruszt,'');
      writeln(wynruszt,'          Vg=',vg:6:4,' m3/s',
      *          Vc=',vc:7:5,' m3/s',          R=',r:4:1)');
      writeln(wynruszt,'          ',rek);
      writeln(wynruszt,'');
      writeln(wynruszt,'');
      end;
poci:wyb:=1;
      a1:=rog/roc;
      bet:=1.118*pot(a1,0.118);
      sa:=vc*roc/rog/vg;
      a22:=sa+0.05;
      y2:=inter(sa,y1,sa1,9);
      a1:=sa*ag/ac;
      c1:=0.538*pot(a1,5.2);
      i:=1;
poci:dz:=dol[i];
      dz:=23*dz;
      j:=1;
      if abs(dz-0.003)<epsi then c:=0.0094 else c:=0.0065;
      y:=c*8*pot(a22,-0.78);
      a:=vg/dz;
      a111:=1.7038*pot(a,0.1112)*pot(roc,0.472)*pot(sa,0.267)
      /pot(rog,0.605)/pot(etg,0.0712)/pot(etg,0.0623);
efi: fi:=fi1[j];
      predkosc;
      d:=1.1288*sqrt(vg/0.95/wa);
      srednica(d1,jd,jk);
      if jk=0 then
          begin

```

```

        writeln(wynruszt,'### dz=',dz:5:3,'    fi=',fi:5:3,
                '### srednica wieksza od',
                ' znormalizowanej');

        goto efik
    end;

    w:=1.273*vg/d/d;
    fi:=0.7854*d*d;
    fo:=fi*f;
    if w>(0.958*wa) then
        begin
            writeln(wynruszt,'dz=',dz:5:3,'    fi=',
                    fi:5:3,'### predkosci',
                    ' wieksza od dozwolonej');

            goto efik
        end;
    if w<=wi then
        begin
            writeln(wynruszt,'dz=',dz:5:3,'    fi=',fi:5:3,
                    '### silny przeciek');

            goto efik
        end;
    if w<(0.858*wa) then
        begin
            writeln(wynruszt,'dz=',dz:5:3,'    fi=',
                    fi:5:3,'### predkosci',
                    ' mniejsza od zalecanej');

            goto efik
        end;

    a2:=dz/al(1);
    ksi:=1.73*sqrt(1-fi)*pot(a2,0.2);
    a2:=sqrt(sa/0.62)*rog/roc/ksi;
    c:=pot(a2,0.333);
    tau:=c/(1+c);

```

```

a2:=1-tau;
a1:=1-bet;
pog:=sqr(w/f1)*ksi*rog*(1-tau*a1)/2/a1/pot(a2,3.0)+2*xi/dz/a1;
hr:=pog/9.81/roc;
if hr<0.024 then
begin
    writeln(wynruzt,'dz=',dz:5:3,' f1=',f1:5:3,
            ' *** wysokość zredukowana',
            ' mniejsza',
            ' od zalecanego minimum');
    goto efik;
end;
pc:=bet*(pog-ksi*rog*tau/2/pot(a2,3));
zp:=hr*f;
wot:=vg/f;
a2:=etg/si;
a1:=rog/(roc-rog);
he:=23.8*ws*pot(a2,0.43)*pot(a1,0.285);
rop:=0.24*pot(se,0.14);
hpi:=(pog-sqr(w/f1)*ksi*rog/2/sqr(1-tau))/9.81/rop/roc;
hi:=hpi+he;
ki:=1;
eh: if h<hi[k] then hi:=hi[k]
    else begin
        if k<ih then
            begin
                ki:=k+1; goto eh;
            end
        else
            begin
                writeln(wynruzt,'dz=',dz:5:3,
                        ' f1=',f1:5:3,
                        ' *** koniec pot');
            end
        end
    end

```

```

'ek większy od',
' stosowanego');

goto efik

endj

endj

a2:=w/hg;
me:=0.4e-5spot(a2,2.56);
ps:=pog-pc;
a2:=d+3.049;
kp:=20.838spot(a2,2.21);
a2:=d+1.524;
kps:=3.857spot(a2,3.45);
dot:=dz/2;
if abs(dot-0.003)<epsi then tot:=0.008 else tot:=2.58dot;
kek:=kp+kps;
sprawbez;
kwi=kryt;
writeln(wynruzt,'');
writeln(wynruzt,' $$$H=',h:4:2,'m', '      D=',d:4:2,'m',
'      F=',f:6:4,'m', '      wg=',w:6:4,'m/s');
writeln(wynruzt,'');
writeln(wynruzt,'      KPO=',kps:5:2,'      KP=',
      kp:4:2,'      KEK=',kek:6:2);
writeln(wynruzt,'      Ek=',ek:6:4, '      eki=',ekl:6:4,
'      KRW=',krw:7:2);
writeln(wynruzt,'');
writeln(wynruzt,' $$$do=',dot:5:3,'m', '      fi=',fi:4:2);
writeln(wynruzt,'');
writeln(wynruzt,'      F      Fm      to',
'      umin      umax');
writeln(wynruzt,'      a2      a2      a',
'      m/s      m/s');
writeln(wynruzt,f:11:4,fel:13:4,tot:13:3,w:13:3,wel:14:3);

```

```

writeln(wynruszt,'          wo          me          zp',
        '          he          hr');
writeln(wynruszt,'          m/s          k6/k6          m3',
        '          m          m');
writeln(wynruszt,wo:11:4,me:13:4,zp:13:3,he:13:3,hr:14:3);
writeln(wynruszt,'          pc          ps          pog',
        '          hpi          rop');
writeln(wynruszt,'          N/m2          N/m2          N/m2',
        '          m');
writeln(wynruszt,pc:11:4,ps:13:4,pog:13:3,hpi:13:3,rop:14:3);
writeln(wynruszt,'!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!',
        '!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!');
writeln(wynruszt,'');
writeln(wynruszt,'');
if (wyb=1) or (wyb=krw) then
    append(wybruszt) else if (wyb>krw) then
        begin
            assign(wybruszt,'c:\polki\wybruszt.pas');
            rewrite(wybruszt);
        if (dan='t') or (bila='t') then
            begin
                writeln(wybruszt,'');
                writeln(wybruszt,'          WYNIKI OBLICZEN POLEK',
                        '          RUSZTOWYCH');
                writeln(wybruszt,'          dla zalozonych parametrow',
                        '          ruchomych (Vg,Vc)');
                writeln(wybruszt,'');
                writeln(wybruszt,'          Vg=',vg:6:4,' m3/s',
                        '          Vc=',vc:7:5,' m3/s');
                writeln(wybruszt,'          ');
                writeln(wybruszt,'');
                writeln(wybruszt,'');
            end
        end
end

```



```

else
begin
    writeln(wybruszt, '');
    writeln(wybruszt, '                WYNIKI OBLICZEN POLEK',
            ' RUSZTOWYCH');
    writeln(wybruszt, '                dla zalozonych parametrow',
            ' ruchowych (Vg,Vc,R)');
    writeln(wybruszt, '');
    writeln(wybruszt, '                Vg=',vg:6:4,' m3/s',
            '                Vc=',vc:7:5,' m3/s',
            '                R=',r:4:1);
    writeln(wybruszt, '                ',rek);
    writeln(wybruszt, '');
    writeln(wybruszt, '');
end;

    end else goto swyb;
    writeln(wybruszt, '');
    writeln(wybruszt, ' $$$H=',h:4:2,' m', '                D=',d:4:2,' m',
            '                F=',f:6:4,' m', '                Wg=',w:6:4,' m/s');
    writeln(wybruszt, '');
    writeln(wybruszt, '                KPO=',kpo:5:2,'                KP=',
            '                kpi:6:2,'                KBK=',ksk:6:2);
    writeln(wybruszt, '                Ek=',ek:6:4,
            '                KRW=',krw:7:2);
    writeln(wybruszt, '');
    writeln(wybruszt, ' $$$do=',dot:5:3,' m', '                fi=',fi:4:2);
    writeln(wybruszt, '');
    writeln(wybruszt, '                F                Fo                to',
            '                wmin                wmax');
    writeln(wybruszt, '                a2                a2                a',
            '                a/s                a/s');
    writeln(wybruszt, f:11:4,fo:13:4,tot:13:3,wi:13:3,wm:14:3);
    writeln(wybruszt, '                wo                oo                zp ',

```

```

      '      ne      hr');
writeIn(wybruszt,'      m/s      k6/k6      m3',
      '      m      m');
writeIn(wybruszt,wo:11:4,me:13:4,zp:13:3,he:13:3,hr:14:3);
writeIn(wybruszt,'      pc      ps      pog',
      '      hpi      rop');
writeIn(wybruszt,'      N/m2      N/m2      N/m2',
      '      m');
writeIn(wybruszt,pc:11:4,ps:13:4,pog:13:3,hpi:13:3,rop:14:3);
writeIn(wybruszt,'!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!',
      '!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!');
writeIn(wybruszt,'');
writeIn(wybruszt,'');
ewyb:if(wyb=1) or (wyb>krw) then wyb:=krw;
efik:j:=j+1;
      if j<ifi then goto efi;
eiki:i:=i+1;
      if i<ib then goto poci;
close(wynruszt);close(wybruszt);
end. ( koniec programu RUSZY )

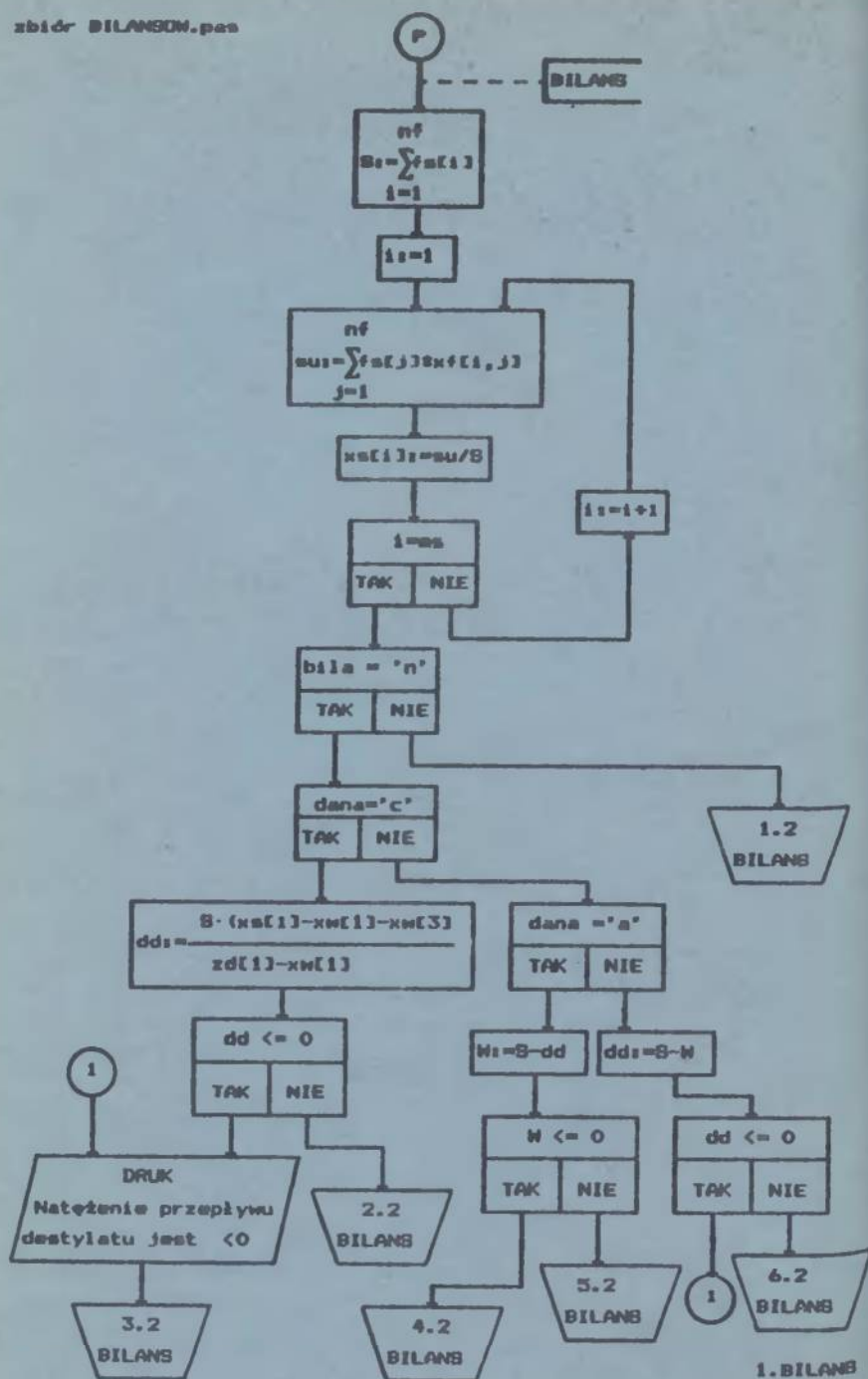
```

7. WYKAZ PROCEDUR I FUNKCJI

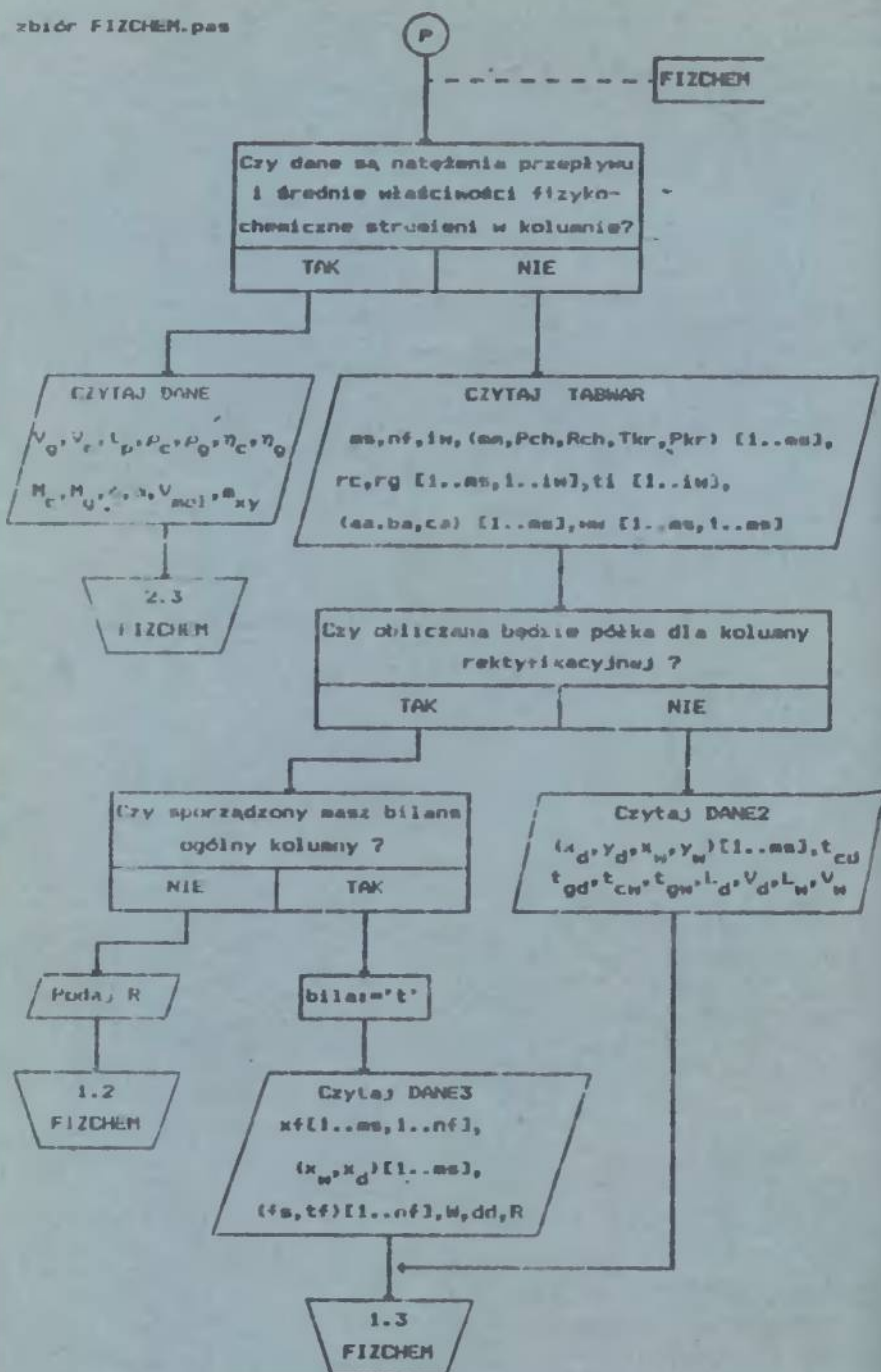
Nazwa zbioru	Nazwa funkcji lub procedury	Specyfikacja	Strona	
			tabulogram	schemat blokowy
BILANGON	bilans	procedura	198	162
FIZCHEM	fizchem	procedura	200	164
GEOSTOC	ror	funkcja	205	167
INTABLIC	intab	procedura	206	168
INTER2	indwa	funkcja	207	169
INTERPOL	inter	funkcja	208	171
KRYTERIA	kryteria	procedura	209	172
KRYTWYB	kryt	funkcja	210	173
LEPCIECZ	ter	funkcja	211	174
LEPGAZ	teta	funkcja	211	175
MABA	mac	funkcja	212	170
NAPPOW	sir	funkcja	212	174
PRZELEW	przel	procedura	213	176
PRZELICZ	przelicz	procedura	215	167
ROWKRO	kro	procedura	215	177
SPRAWNO	sprawnosc	procedura	216	179
TEMPERAT	temp	procedura	218	181
ULWAGOME	ulwag	procedura	221	186
WLASNOBC	wlas	procedura	222	188
WLFICH	wlasfiz	procedura	222	189
WYBRED	srednica	procedura	226	192
	fmax	funkcja	ZAWOR	170
	gradcis	funkcja	KOLP	193
	predkosc	procedura	RUSZT, BITBEZ	195
	sprawbez	procedura	RUSZT, BITBEZ	196
	zwieksz	procedura	ZAWOR	195

8. SCHEMATY BLOKOWE PROCEDUR

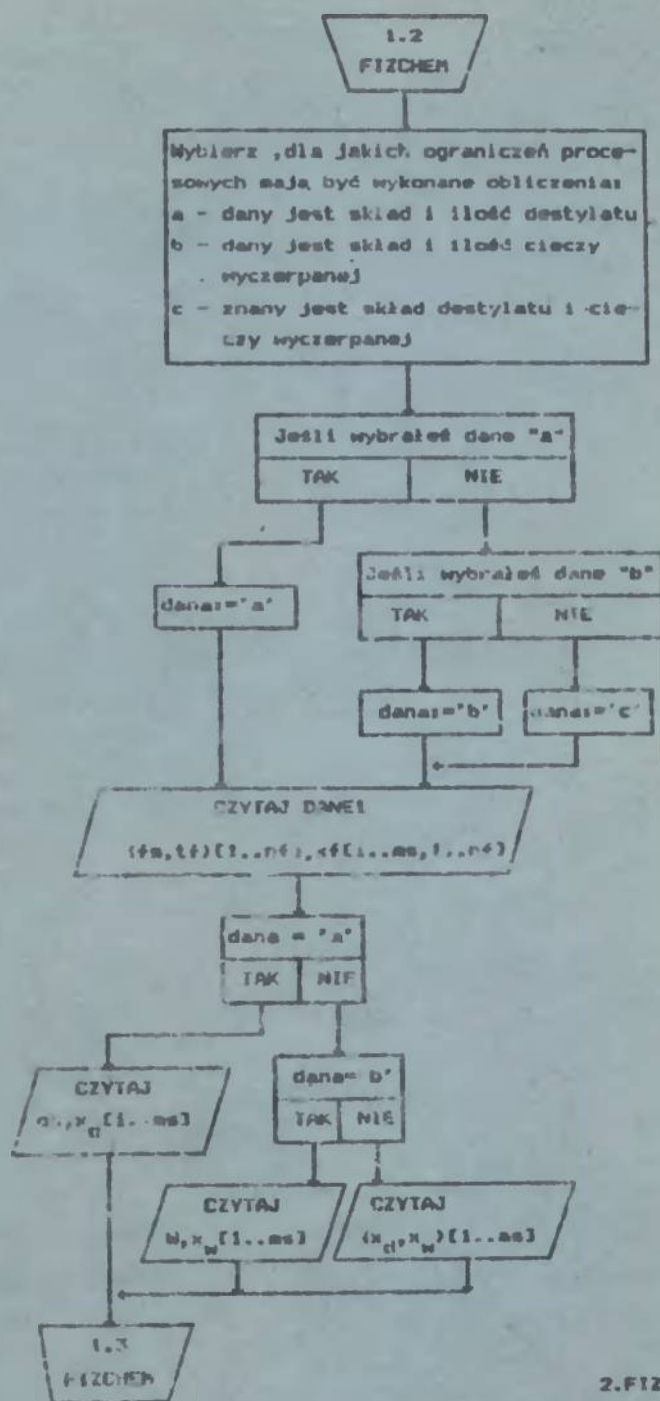
zbior BILANSOW.pas

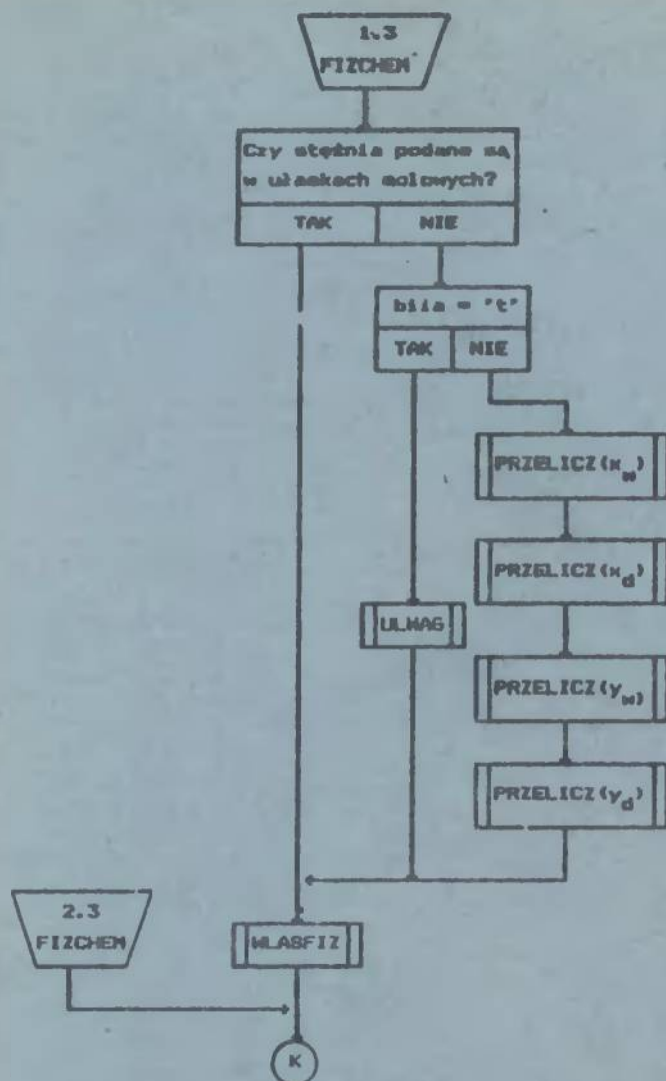


zbiór FIZCHEM.pas



1. FIZCHEM



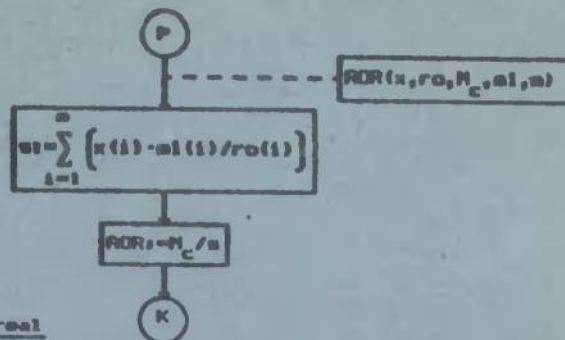


procedure FIZCHEM (bezparametrowa)

(czytanie lub obliczanie (w zależności od danych) właściwości
fizykochemicznych strumieni cieczy i par w kolumnie)

3.FIZCHEM

zbiór SCST08C.pas

function ROR - typ real(Obliczanie gęstości mieszaniny cieczy kg/m^3)

parametry formalne - przekazywane przez wartości:

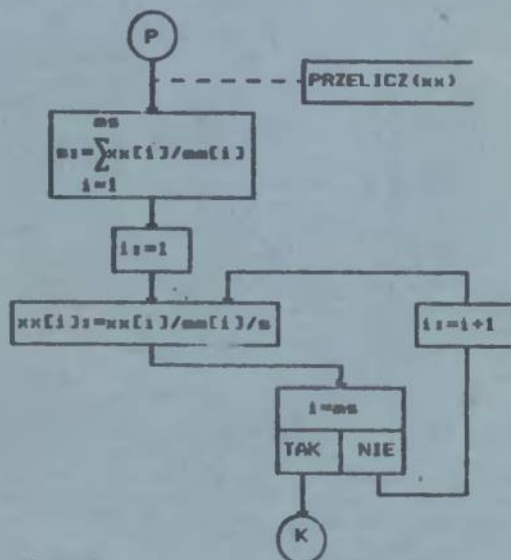
real array x[1..n], real a1[1..n], real ro[1..n], real M_c, integer nzmienna lokalna: real w1; integer i

1. ROR

```

      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  
```

zbiór PRZELICZ.pas



procedura PRZELICZ

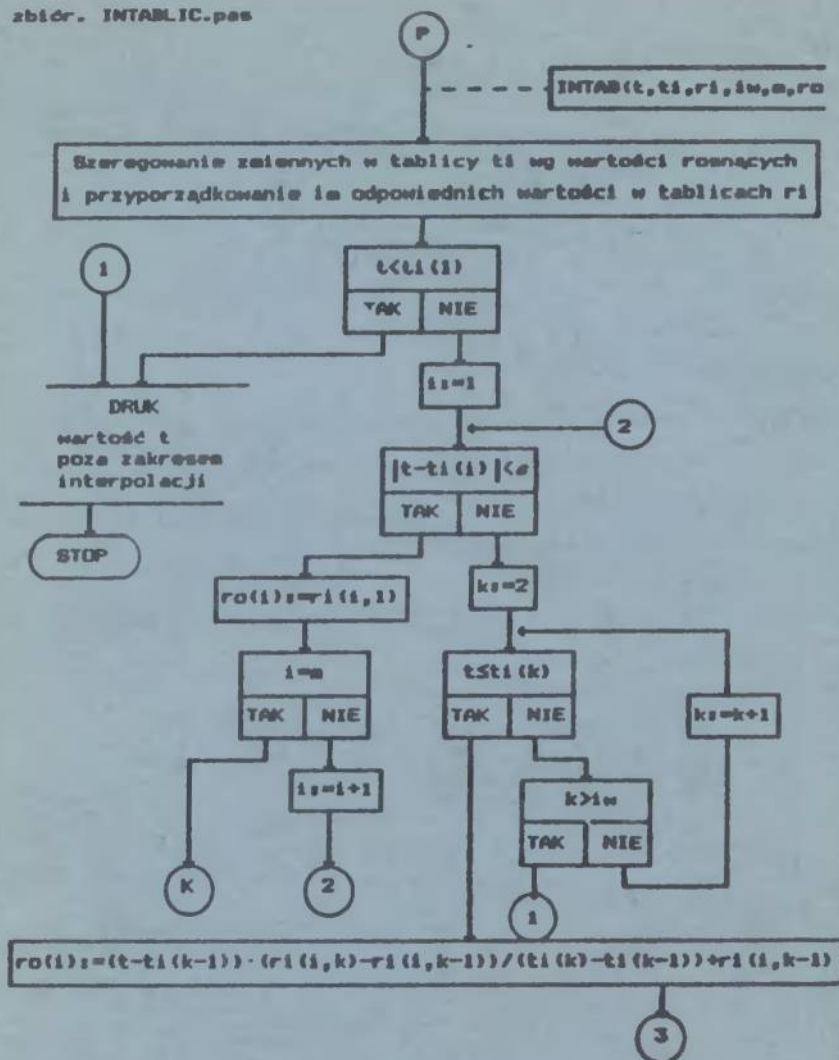
(przeliczanie stężeń wyrażonych

w ułamkach wagowych na stężenia wyrażone w ułamkach molowych)

parametr formalny przekazywany przez zmienną: real array xx[1..ms]zmienna lokalna: integer i; real w

1. PRZELICZ

zbiór. INTABLIC.pas



procedure INTAB (interpolacja z dwóch tablic)

parametry formalne przekazywane przez wartości:

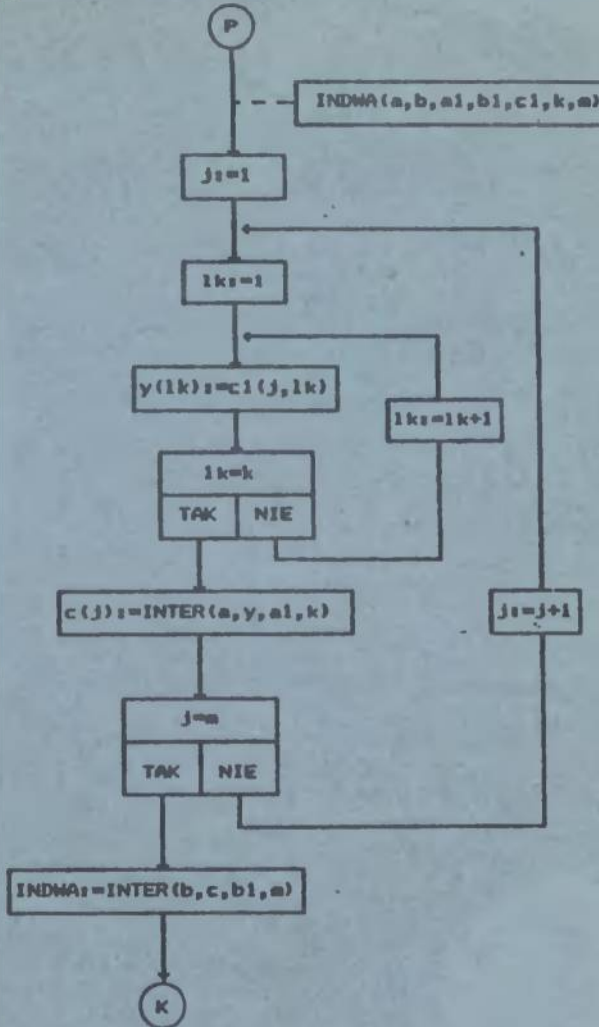
real t; integer iw, m; real array ti[1..iw], ri[1..m, 1..iw]

parametr formalny przekazywany przez zmienną:

real array ro[1..m]

zmiennne lokalne: integer i, k

zbiór INTER2.pas



function INDWA—typ real

(interpolacja funkcji dwóch zmiennych)

parametry formalne przekazywane przez wartości

real array a1[1..k], b1[1..m], c1[1..m, 1..k]

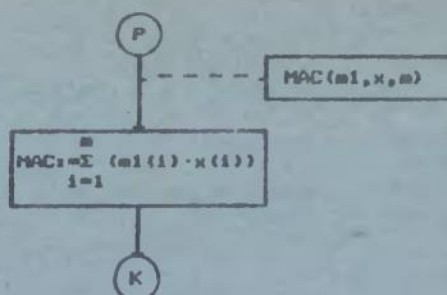
integer k, m ; real a, b ; real function INTER

parametry lokalne

integer j, lk ; real array c[1..m], y[1..k]

1. INDWA

zbiór MASA.pas



function MAC-typ real

(obliczanie addytywnych właściwości fizykochemicznych (masy cząsteczkowej kg/kmol, gęstości mieszaniny par kg/m³))

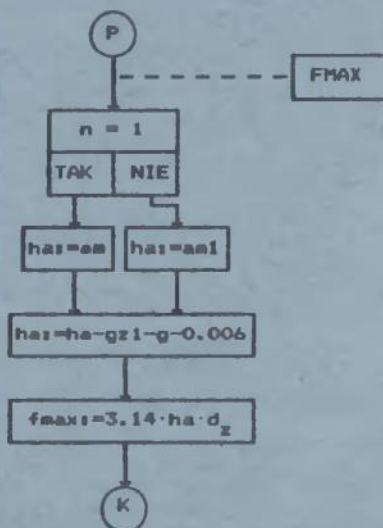
parametry formalne przekazywane przez wartość :

real array x[1..m], m1[1..m] ; integer m

zmienna lokalna: integer i

1.MAC

(w zbiorze ZAWOR)

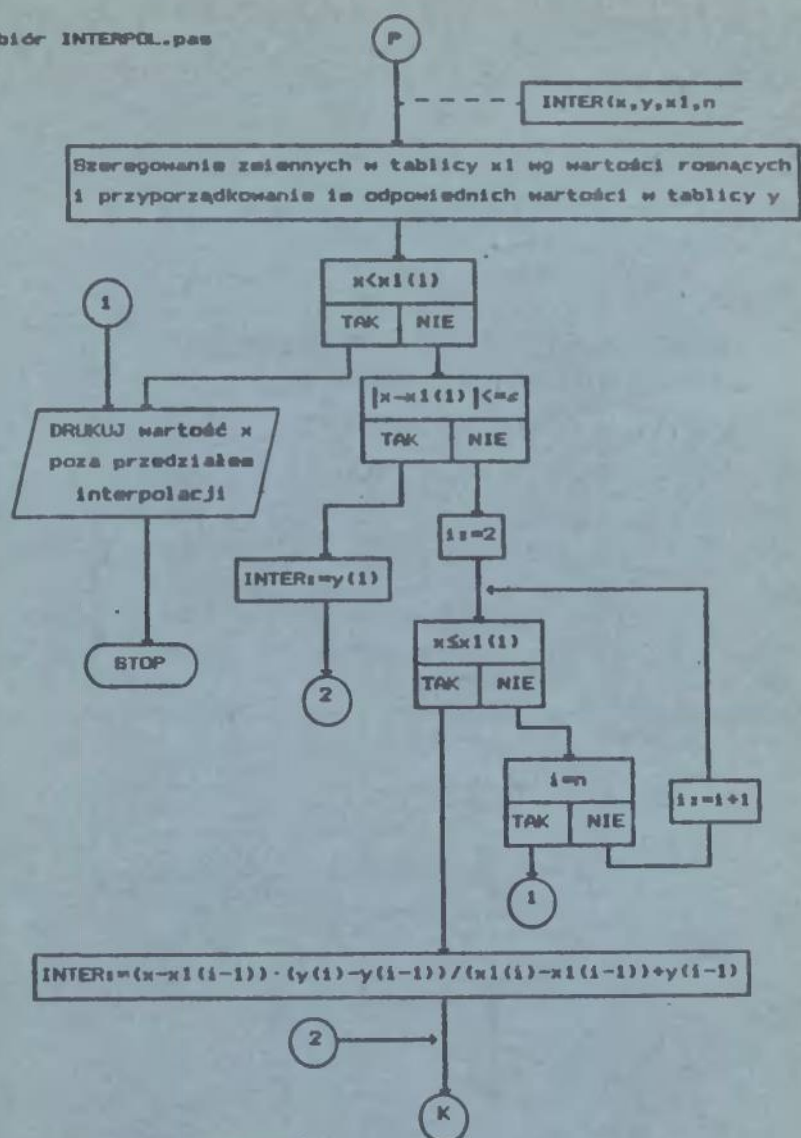


function FMAX - typ real

(Obliczanie powierzchni dla przepływu gazu przy maksymalnym podniesieniu zaworka)

1.FMAX

zbiór INTERPOL.pas



function INTER - typ real

(interpolacja funkcji jednej zmiennej)

parametry formalne przekazywane przez wartość

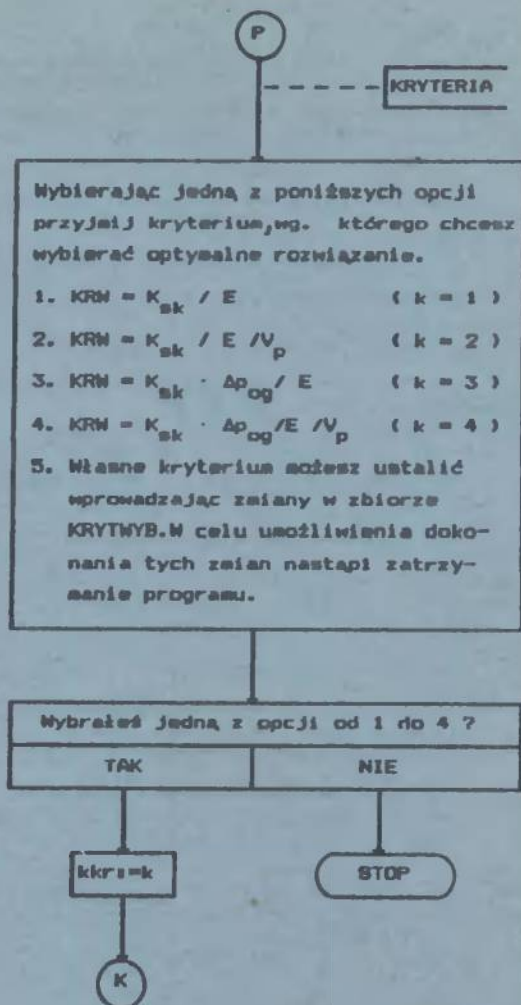
real x ; integer n ; real array y[1..n], x1[1..n]

zmienna lokalna:

integer i

1. INTER

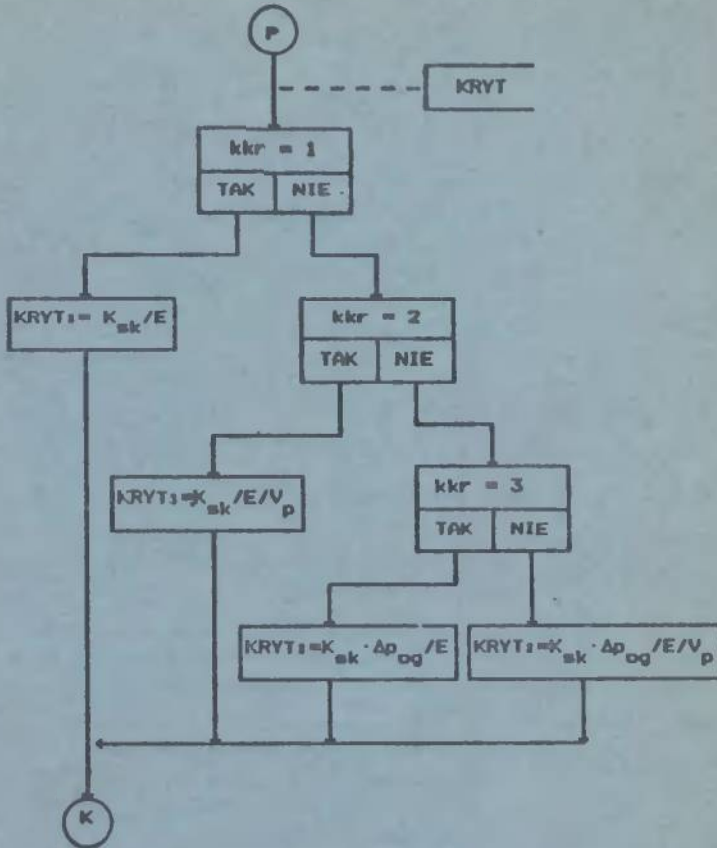
zbiór KRYTERIA.pas

procedura KRYTERIA - bezparametrowa

(Ustalanie kryterium wg., którego będzie wybierane rozwiązanie optymalne).

1.KRYTERIA

zbiór KRYTWYB.pas

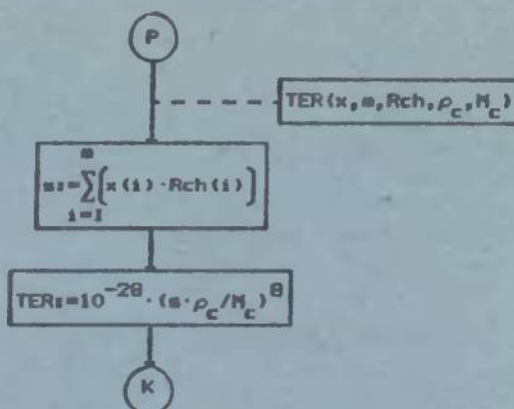


funkcja KRYT - bezparametrowa

(Obliczanie wartości kryterium wyboru w zależności od opcji
przyjętej w procedurze KRYTERIA)

1.KRYTWYB

zbiór LEPCIECZ.pas

function TER - typ real(obliczanie lepkości mieszaniny cieczy (Ns/m²))

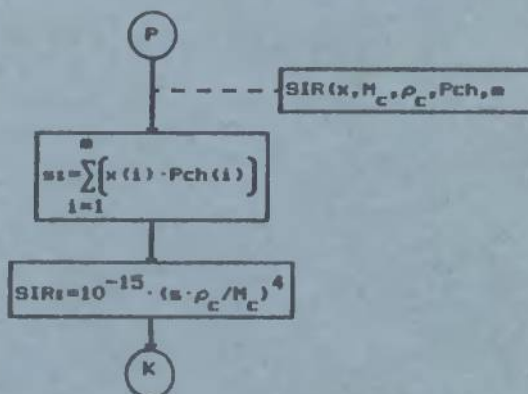
parametry formalne przekazywane przez wartości:

real ρ_c, M_c ; real array x[1..m], Rch[1..m]; integer mzmiennne lokalne: real s; integer i

1. TER

* * * * *

zbiór NAPPON.pas

function SIR - typ real

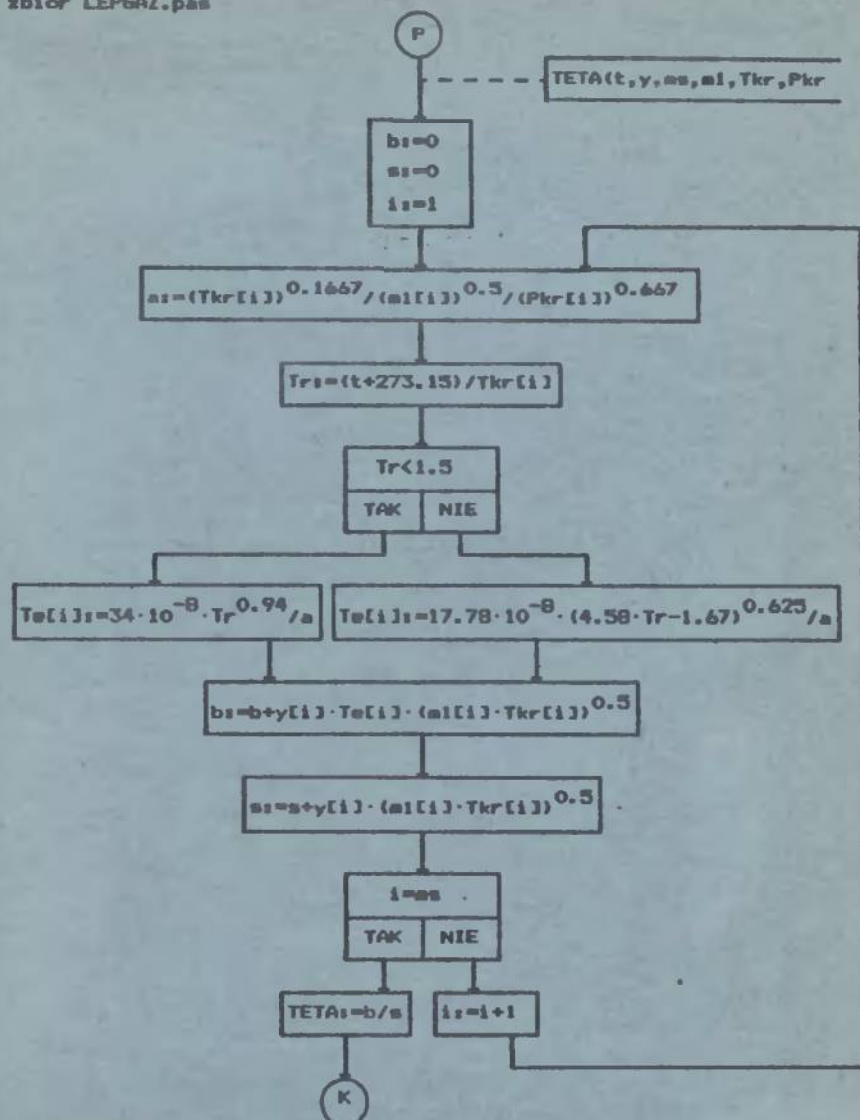
(obliczanie napięcia powierzchniowego mieszaniny cieczy (N/m))

parametry formalne przekazywane przez wartości:

real ρ_c, M_c ; real array x[1..m], Pch[1..m]; integer mzmiennne lokalne: real s; integer i

1. SIR

zbiór LEPAZ.pas



function TETA-typ real

(obliczanie lepkości mieszaniny gazów)

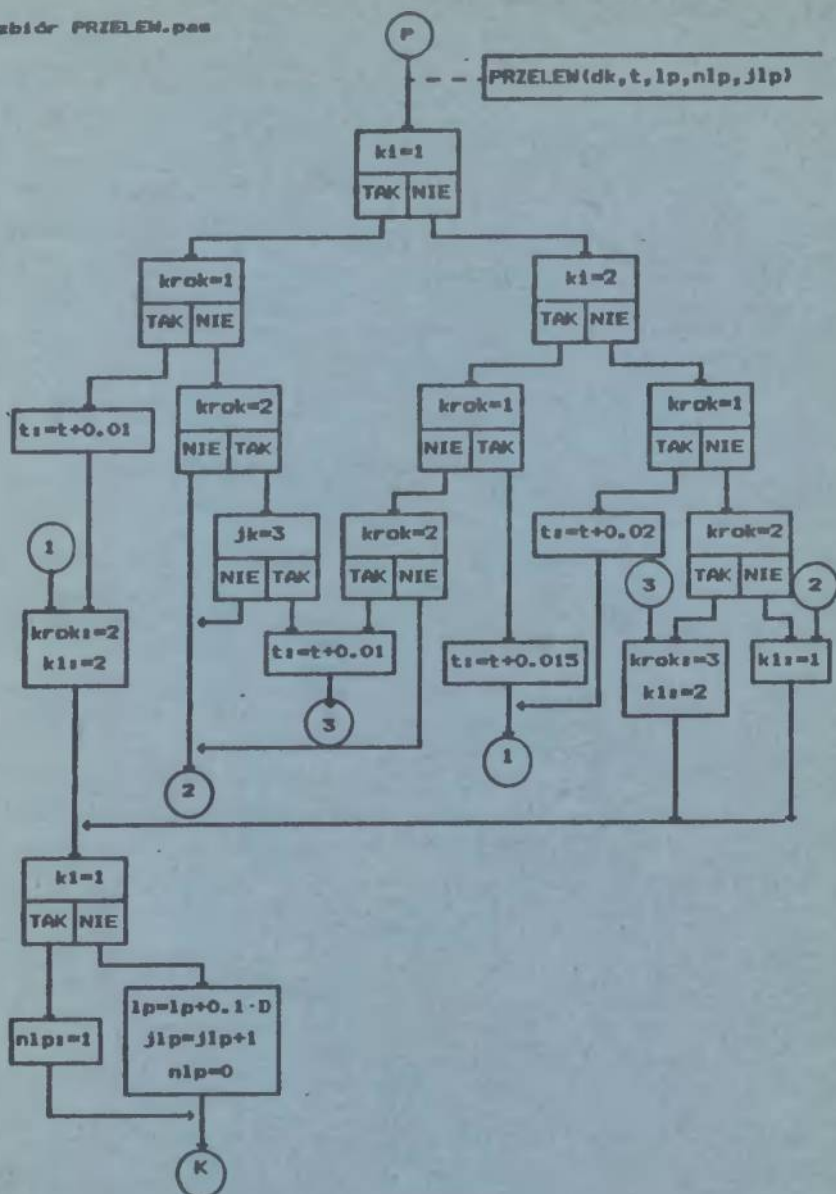
parametry formalne przekazywane przez wartości

real i, real array y, Tkr, Pkr, ml[1..ms]; integer ms

parametry lokalne: integer i, real a, b, s, tr; real array te[1..ms]

1.TETA

zbiór PRZELEN.pas



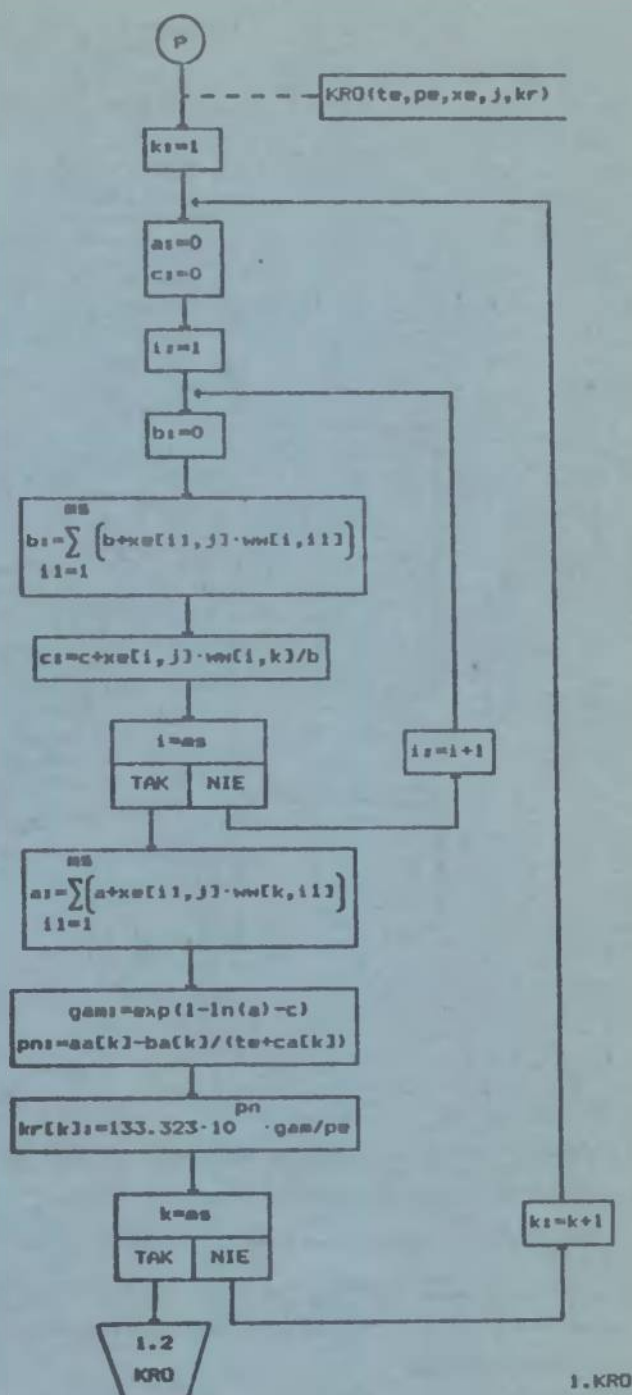
procedure PRZELEN

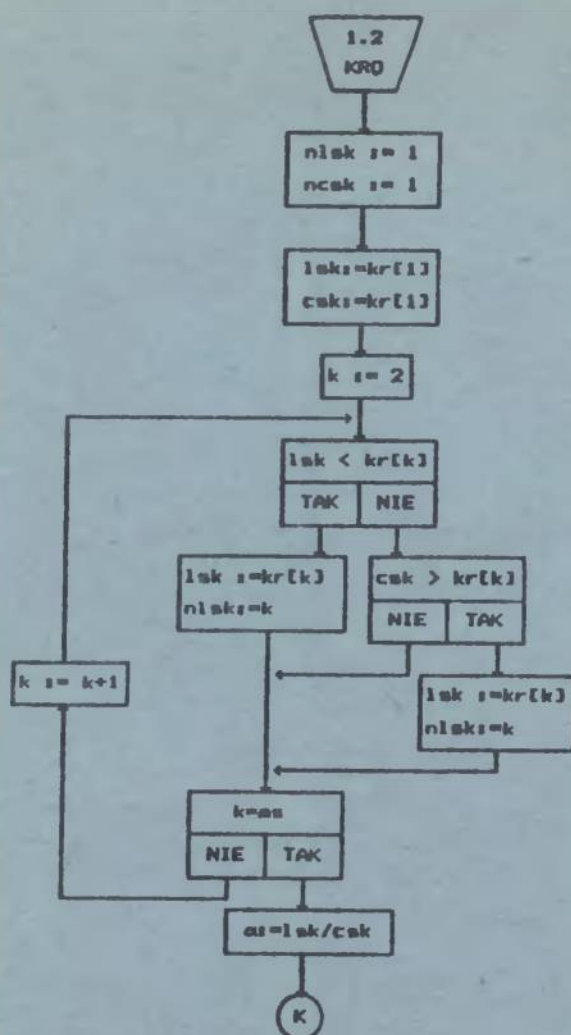
{ustalenie wartości zmiennej pomocniczej nlp przy zwiększaniu lp}

Parametry przekazywane przez zmienną: real t,lp; integer nlp,jlpParametry przekazywane przez wartość: real dk

1.PRZELEN

zbiór ROMKRO.pas





procedure KRO

(obliczanie współczynników równowagi fazowej jako funkcji temperatury, ciśnienia i składu oraz lotności względnej)

parametry formalne przekazywane przez wartość

integer j; real te, pe; real array xel[1..ms, 1..n]

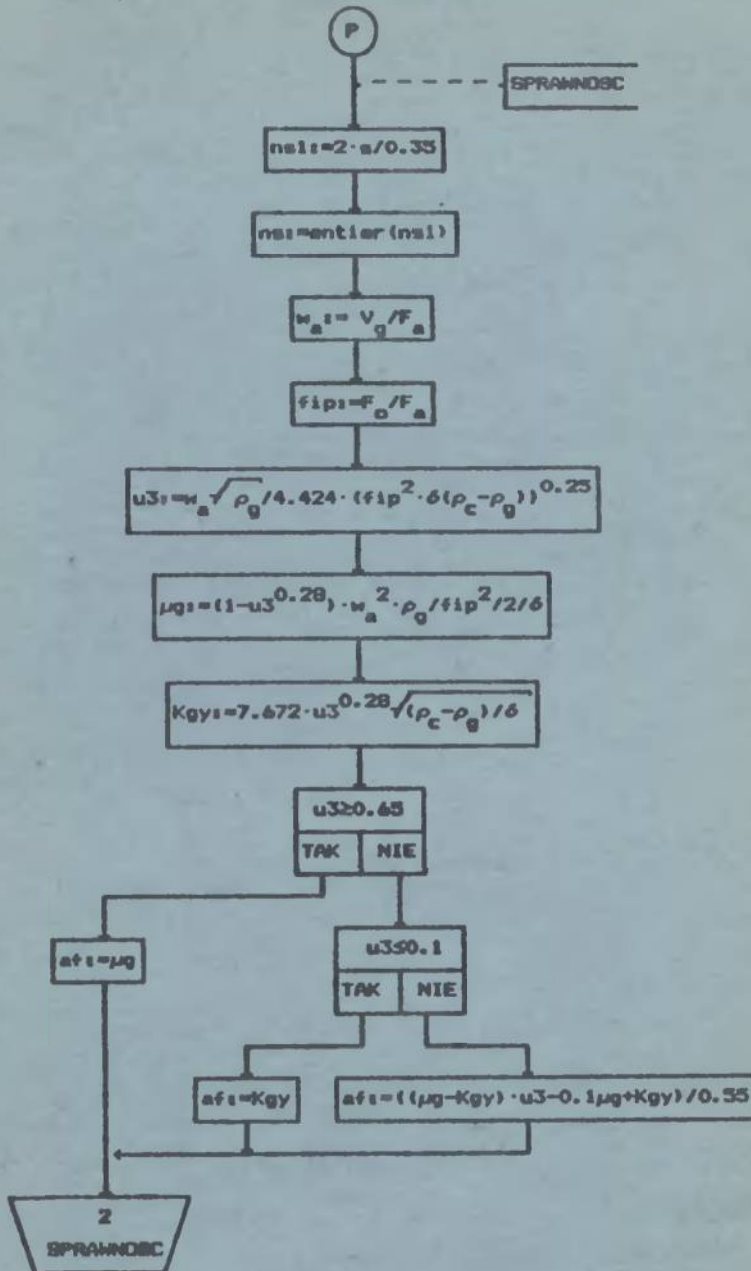
parametry formalne przekazywane przez zmienną: array kr[1..ms]

zmiennne lokalne:

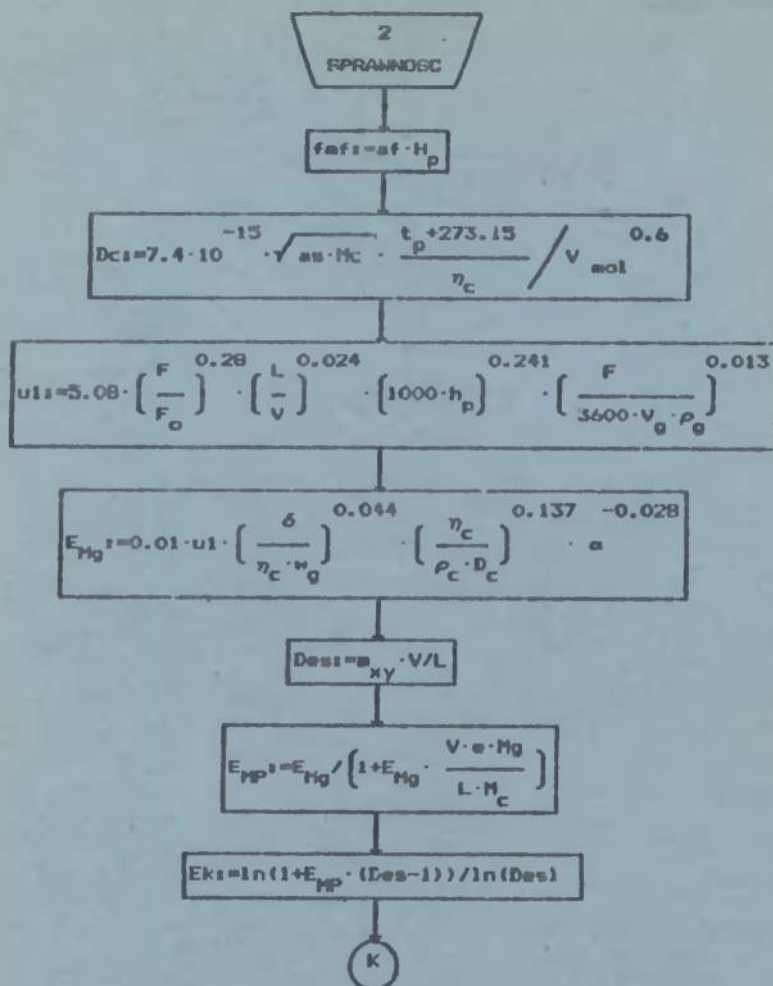
integer i, j, k; real a, b, c, pn, gam

2. KRO

zbiór SPRAWNO.pas



1. SPRAWNO



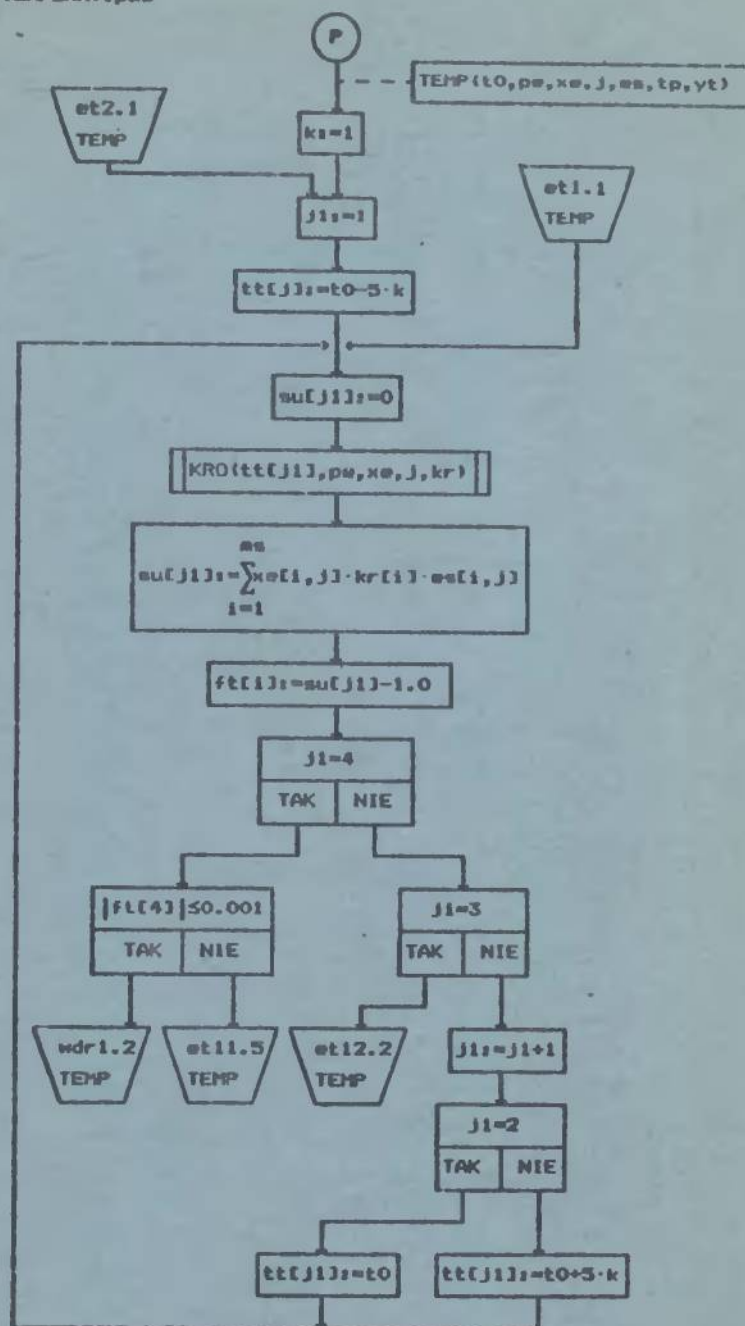
procedure SPRAWNOSC (bezpparametrowa)

(procedura do obliczenia sprawności kolumny z półkami przelewowymi i powierzchni międzyfazowej na tych półkach).

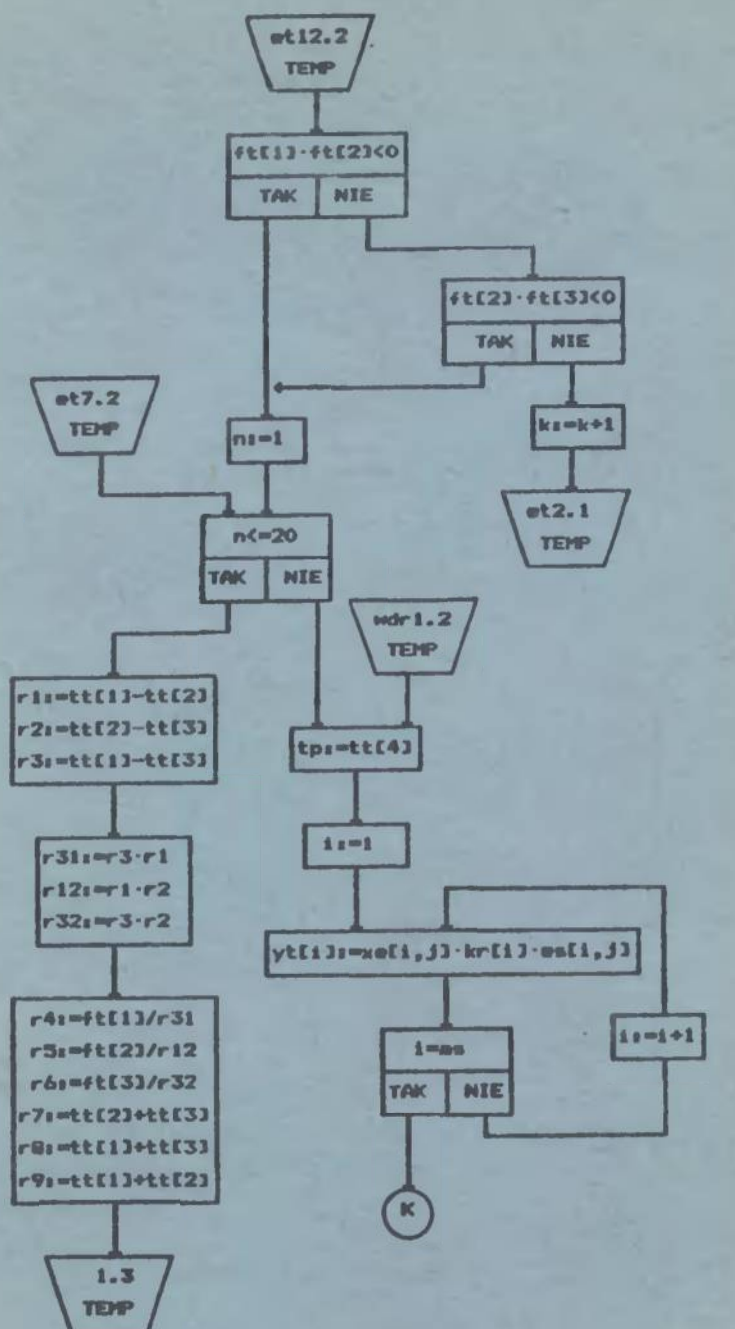
zmiennne lokalne:

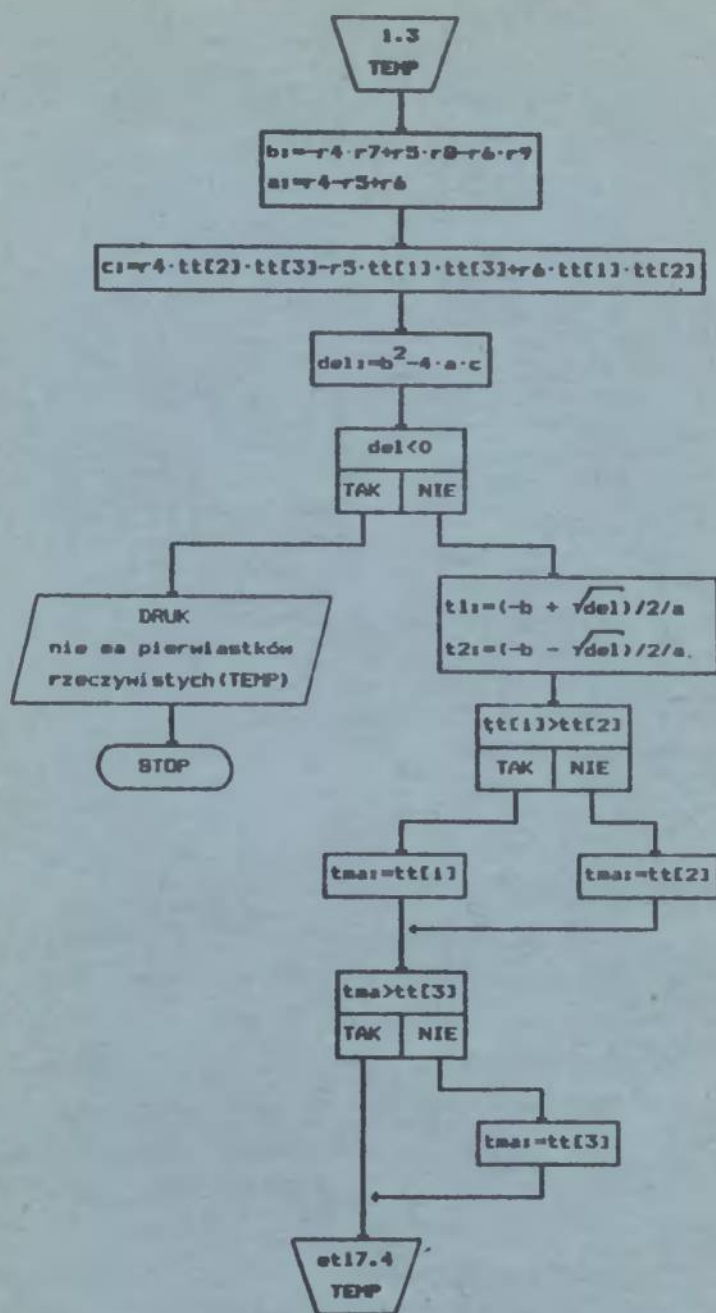
real, fip, u3, af

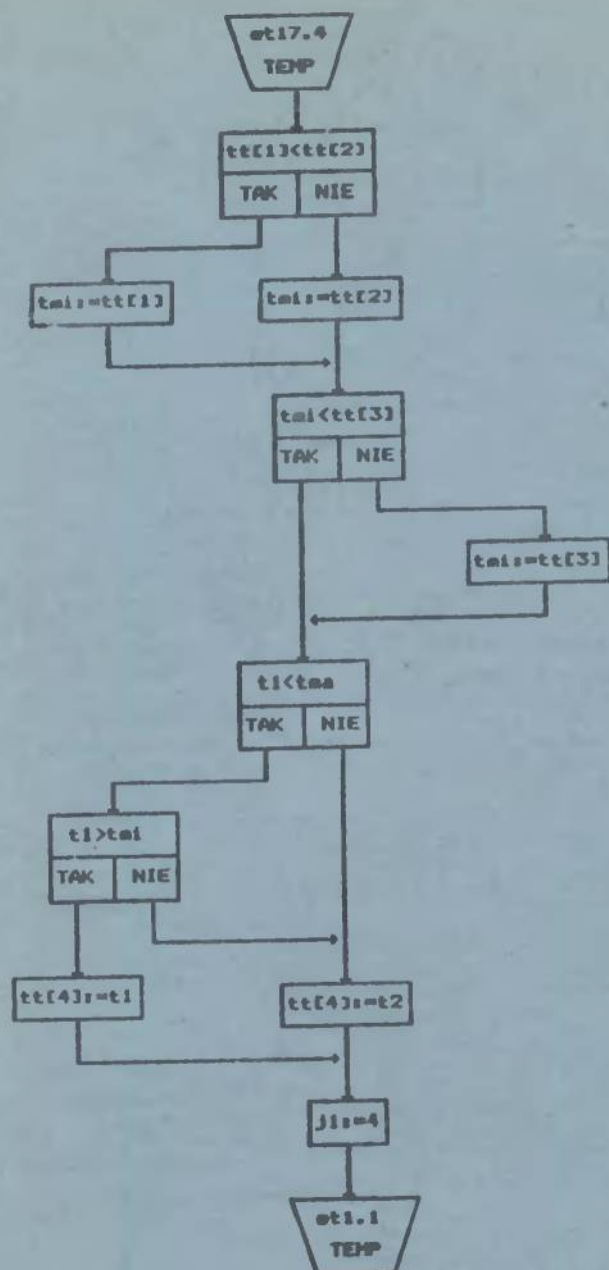
zbiór TEMPERAT.pas

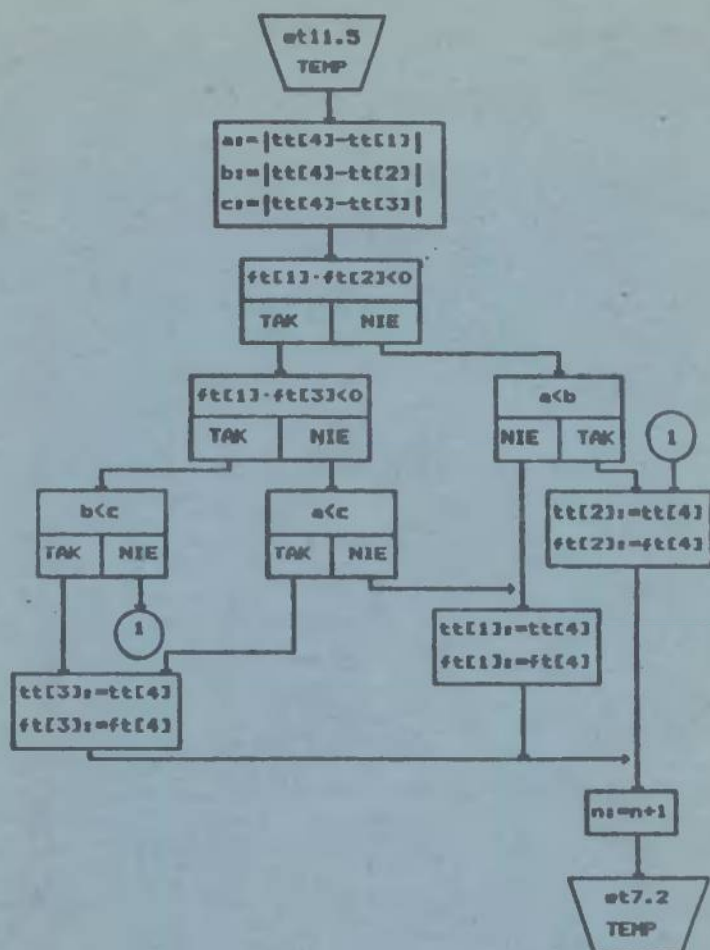


1. TEMP









Procedure TEMP

(obliczanie temperatury wrzenia cieczy o zadanych składzie oraz składu par pozostających w równowadze z tą cieczą)

Parametry formalne przekazywane przez wartości:

real t0, p0; integer j; real array (xe, es)[1..es, 1..j]

Parametry formalne przekazywane przez zmienną:

real tp; real array yt[1..es]

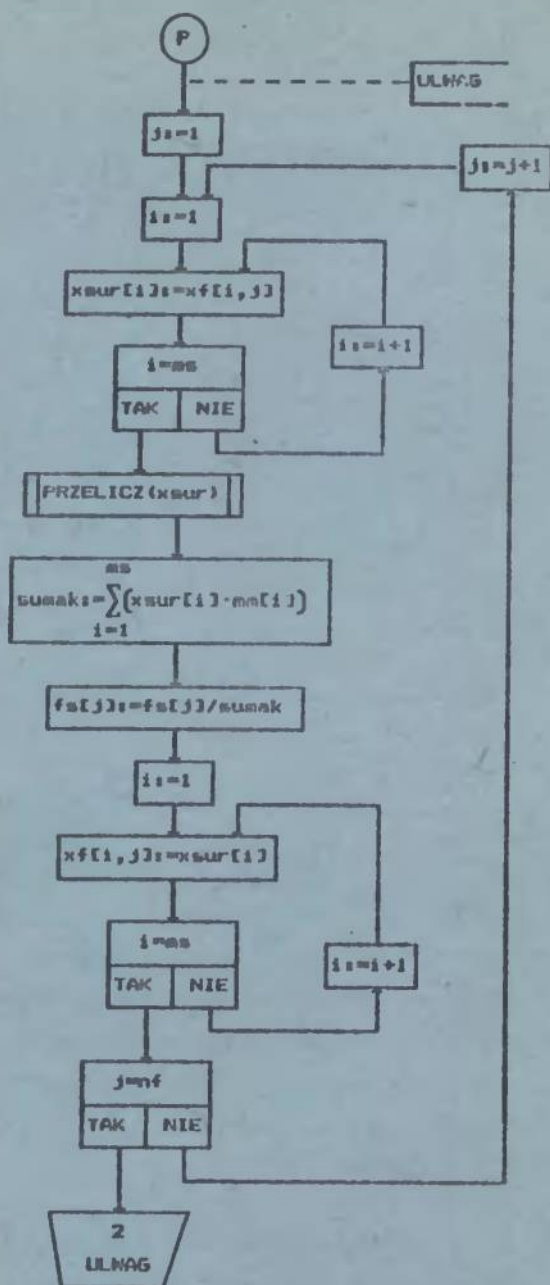
zmiennne lokalne:

integer k, i, j1, ni; real a, b, c, del, r1, r2, r3, r4, r5, r6, r7, r8, r9,

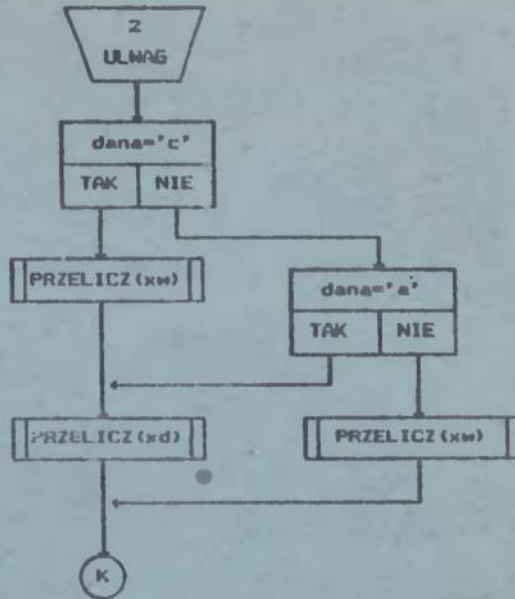
r31, r32, r12, t1, t2, t0a, t0i; real array (tt, ft, su)[1..4]

5. TEMP

zbiór ULNAGONE.pas



1. ULNAG



procedure ULWAG

{przeliczanie udziałów wagowych na udziały molowe oraz natężenia
 strówki z kg/h na kmol/h}

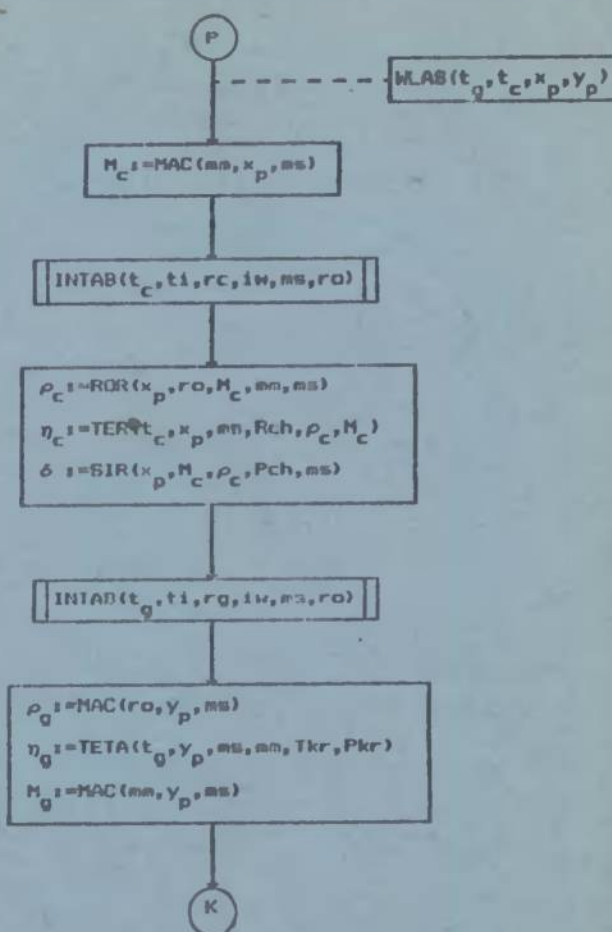
zmienne lokalne:

integer i,j

real sumak

real array xsur[1..ms]

zbiór WLASNOSC.pas



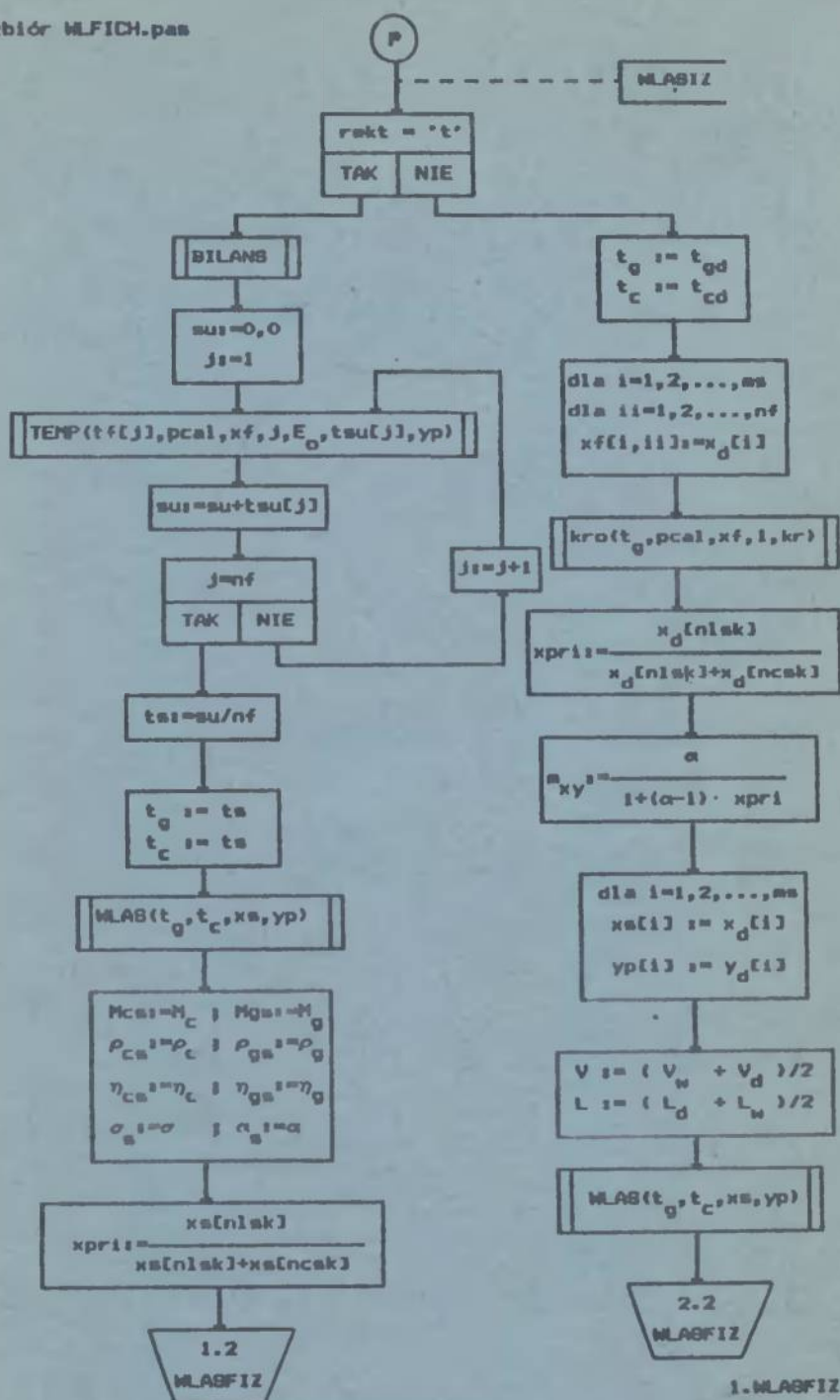
procedure WLAS

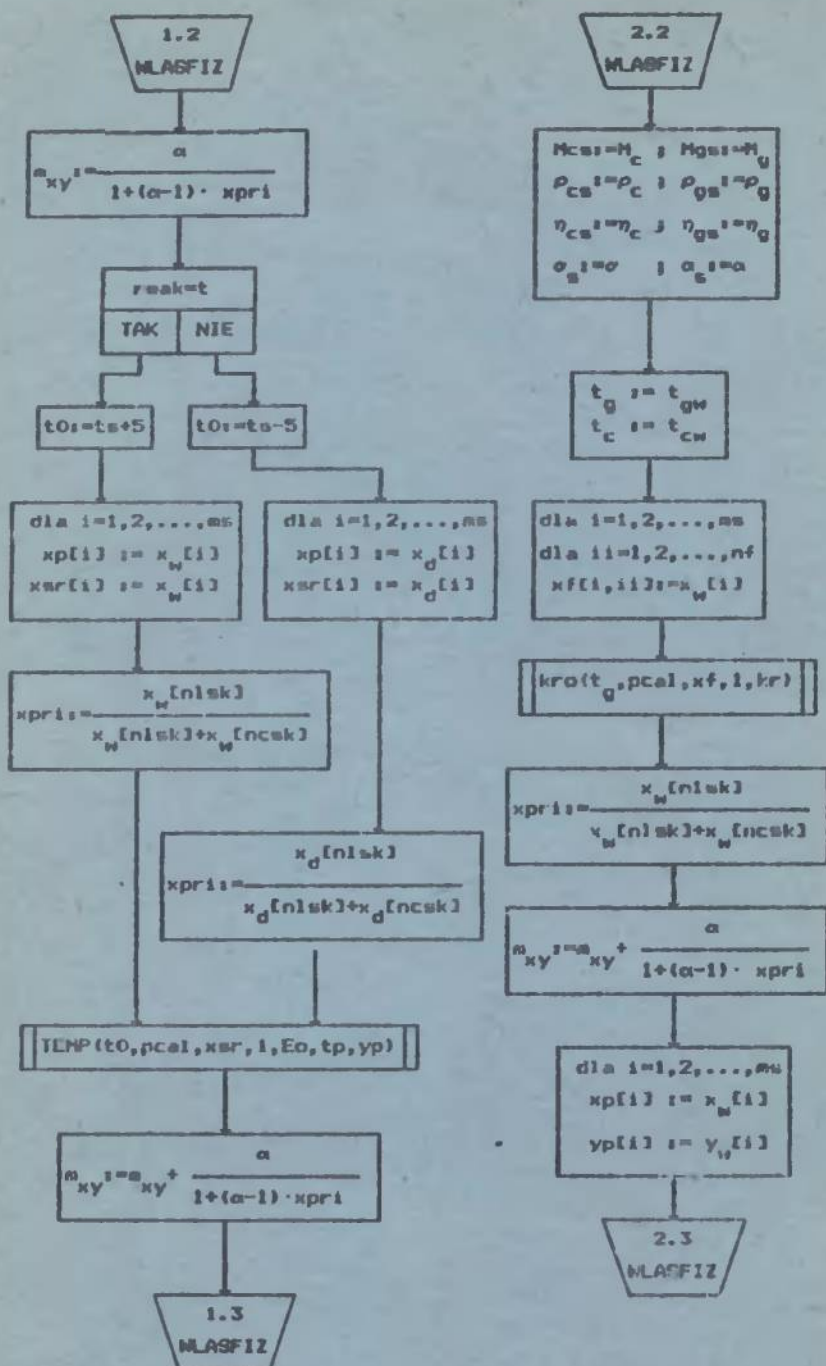
(obliczanie właściwości fizykochemicznych mieszanin cieczy i par)
parametry formalne przekazywane przez wartości:

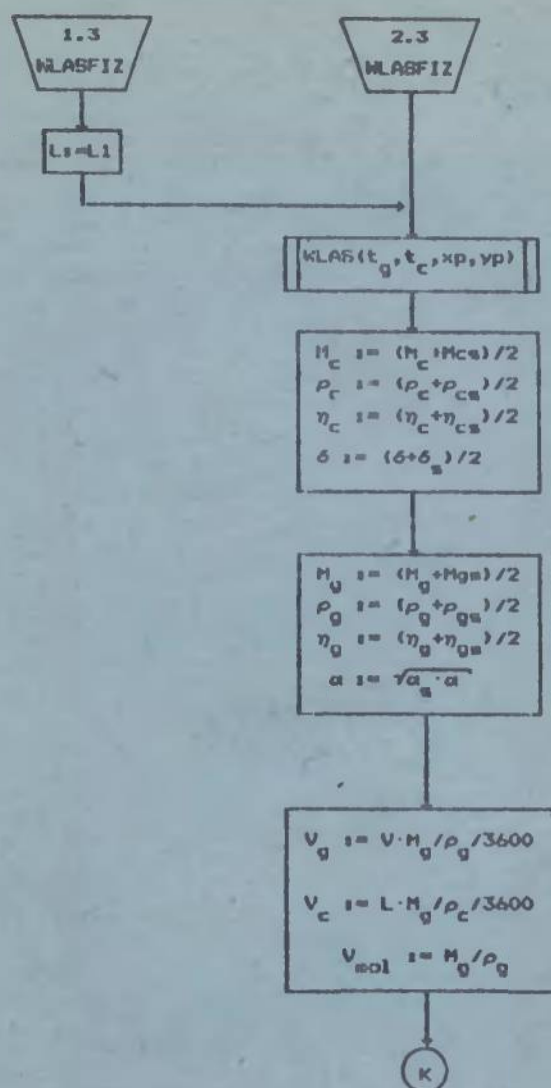
real t_c, t_g

array $(x_p, y_p)[1..ms]$

zbiór WLFICH.pas







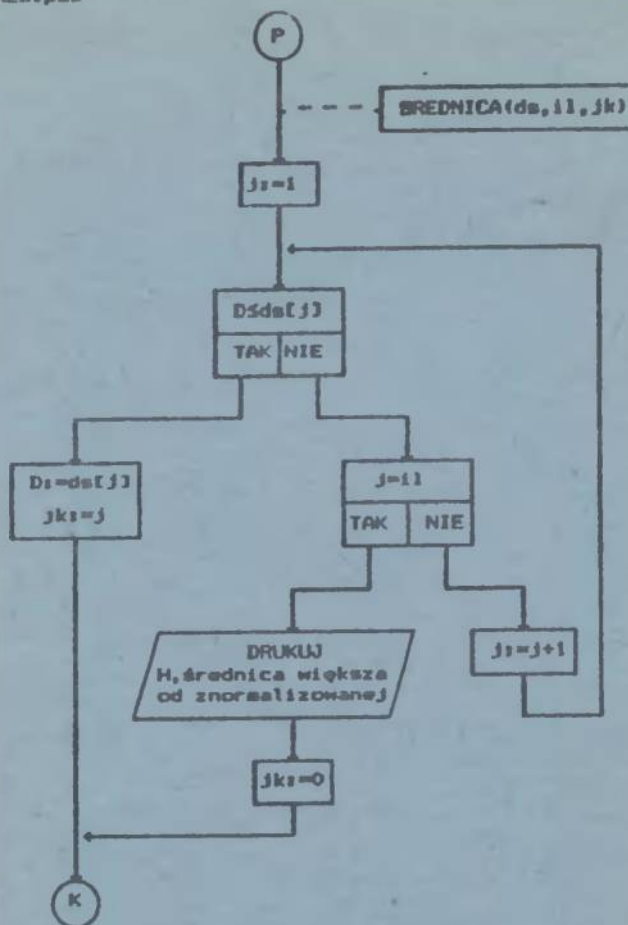
procedure WLABFIZ

(obliczenie średnich parametrów fizykochemicznych strumieni cieczy i par w dolnej lub górnej części kolumny)

zmienna lokalna integer J, ii real su

3.WLABFIZ

źródło WYBRED.pas

Procedure BREDNICA

(procedura wybierająca średnicę z tablicy dla ustalonej

wg normy BN-85/2201-05, określenie indeksu tej średnicy - jk)

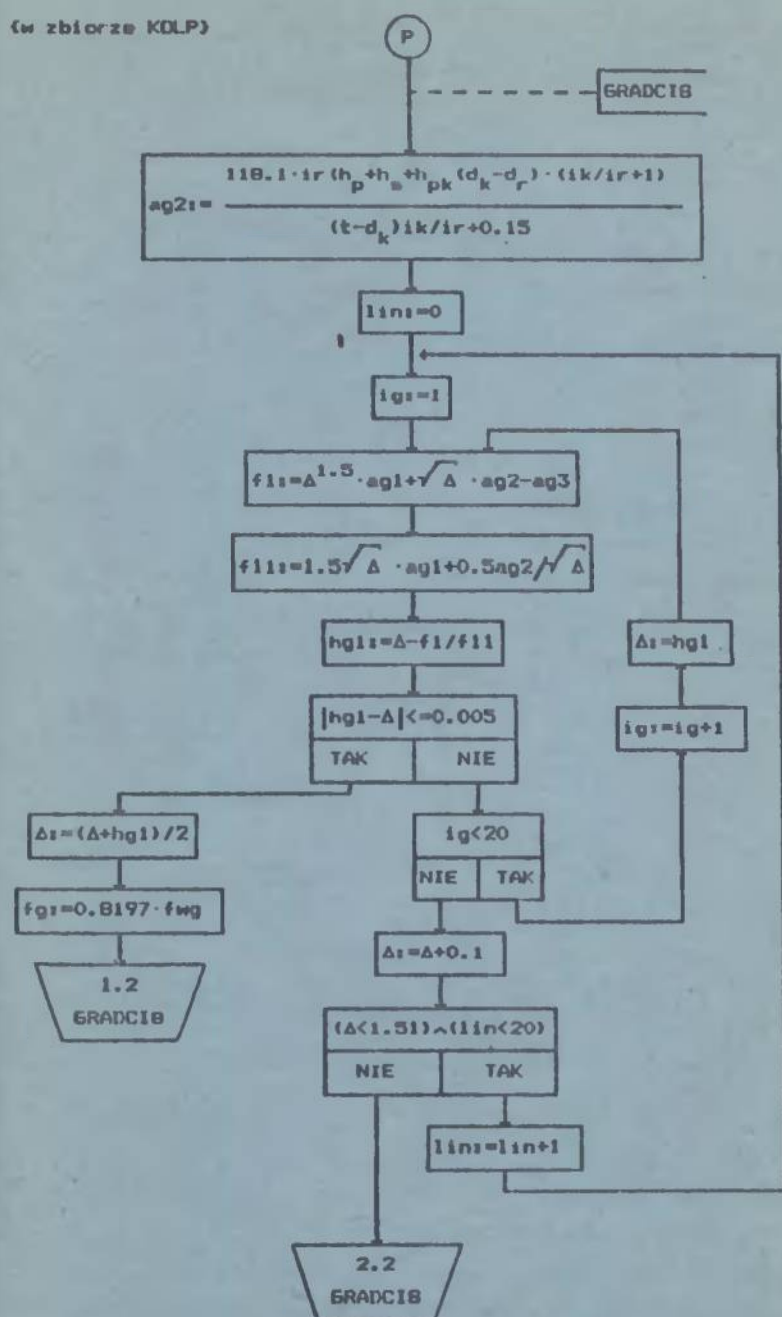
Parametry formalne przekazywane przez wartości:

real array ds ; integer i1

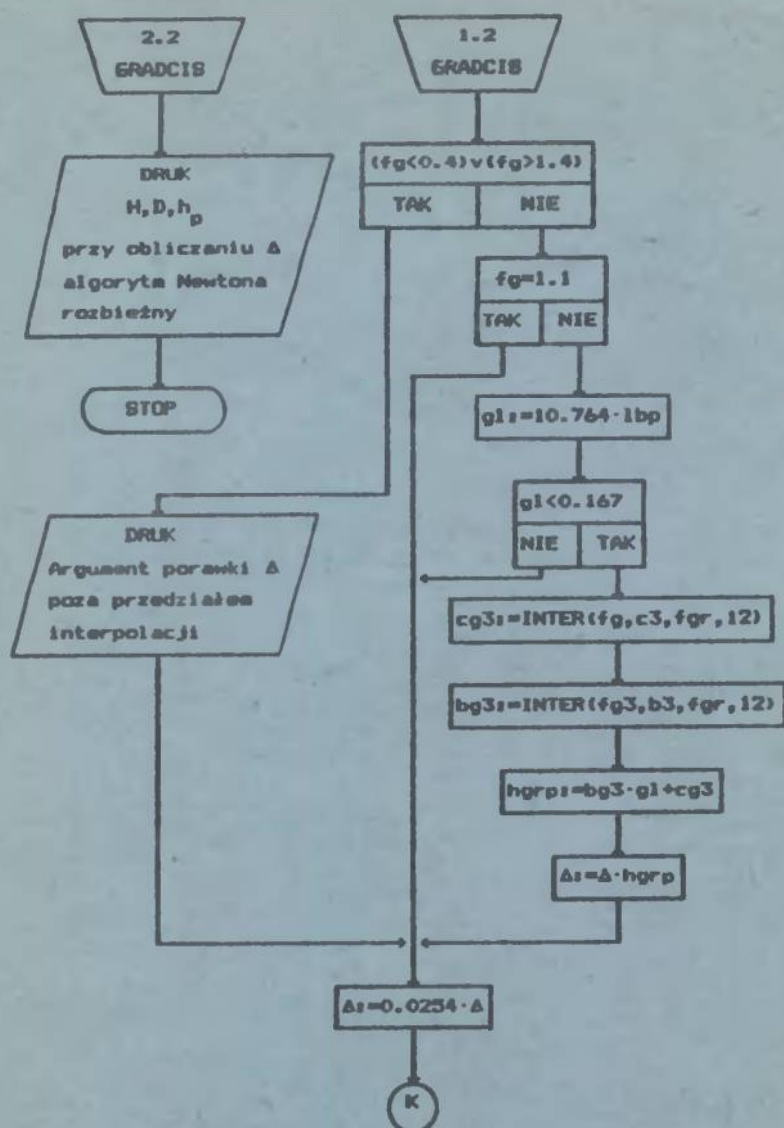
Parametry przekazywane przez zmienną:

integer jkZmienna lokalna: integer j

(w zbiorze KOLP)



1.6GRADCIB



GRADCIS - procedura bezparametrowa

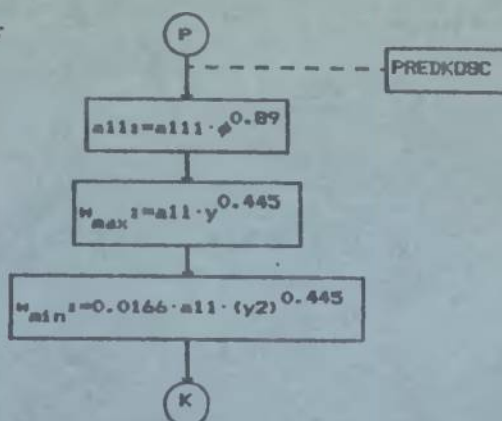
(obliczanie gradientu ciśnienia)

zmiennne lokalne: integer lin, ig

real ag2, bg3, cg3, fg, fi, fi1, g1, hgrp.

2. GRADCIS

(w zbiorach RUSZT
i SITBEZ)



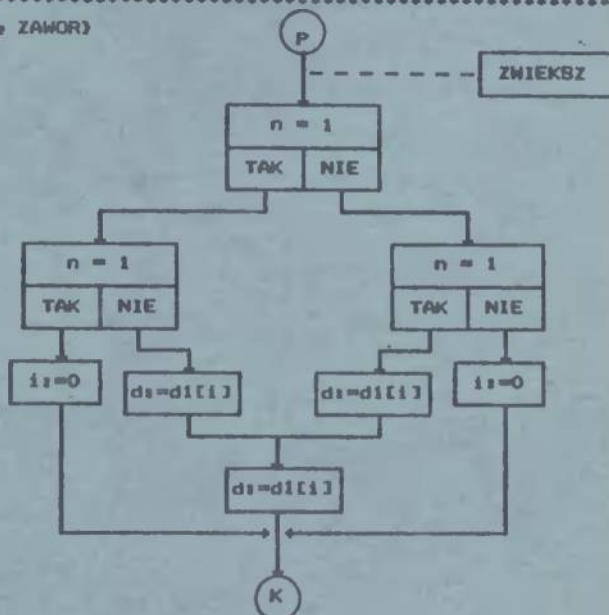
procedure PREDKOSC — bezparametrowa

(Obliczanie minimalnej i maksymalnej prędkości gazu w kolumnie z półkami rusztowymi i sitowymi bez elementów spływowych)

1. PREDKOSC

=====

(w zbiorze ZAWOR)

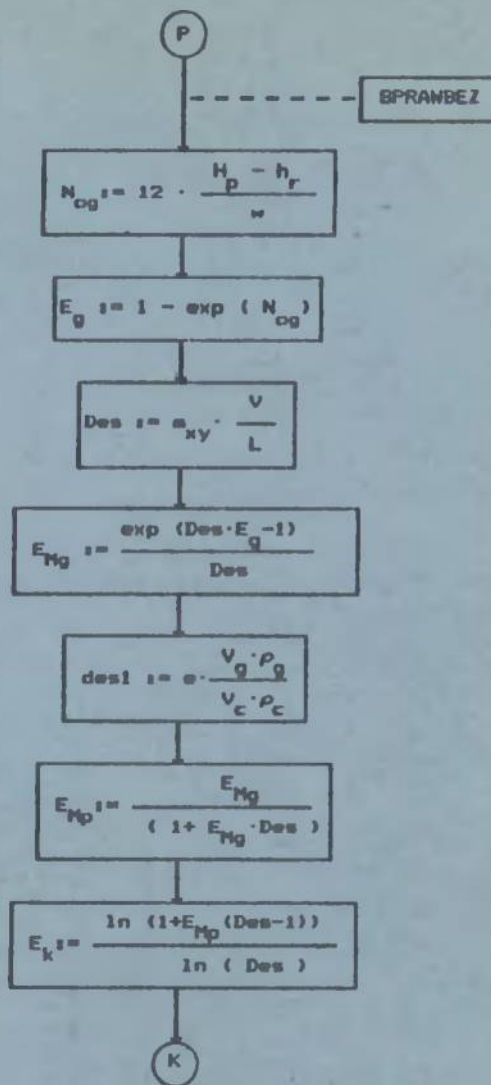


procedure ZWIEKSZ — bezparametrowa

(Ustalanie licznika tablicy d1 lub d2 przy zwiększaniu średnicy półki zaworkowej)

1. ZWIEKSZ

(w zbiorach RUSZT
i BITBEZ)



procedure SPRAWBEZ (bezparametrowa)

(obliczanie sprawności kolumny z półkami bezprzelowowymi)

parametry lokalne :

real : a, N_{cg}, E_g, des1, E_{Mg}, E_{Mp}

1. SPRAWBEZ

9. TABULOGRAMY PROCEDUR

(zbior BILANSOW .pas)

```

procedure bilans;
  label kend;
  var i,j :integer;
      su,e1,e2,e3,e4,a1,b1,c1,a2,b2,c2:real;
  begin
    s:=0;
    for i:=1 to nf do s:=s+f[i];
    for i:=1 to es do
      begin
        su:=0;
        for j:=1 to nf do
          su:=su+f[j]*x[i,j];
        xs[i]:=su/s;
      end;
    if i = 'n' then
      begin
        case dana of
          'a':begin
            w:=s-dd;
            if w <=0 then
              begin
                writeln('Natezenie cieczy wyczerpanej jest',
                  ' ujemne w=',w);
                writeln('NALEZY SKORYGOWAC DANE W ZDIORE DANE1');
                halt;
              end;
            for i:=1 to es do
              begin
                xw[i]:=(s*xs[i]-dd*xd[i])/w;
                if xw[i]<=0 then
                  begin

```

```

        writeln('Stężenie składnika o nr.',i,'w cieczy',
                ' wyczerpanej jest ujeane.');
```

writeln('NALEŻY SKORYGOWAC DANE W ZBIORZE DANE1');

halt;

end

end;

end;

'b':begin

dd:=a-w;

if dd <=0 then

begin

writeln('Natężenie przepływu destylatu jest',

' ujeane');

writeln('NALEŻY SKORYGOWAC DANE W ZBIORZE DANE1');

halt;

end;

for i:=1 to m do

begin

xd[i]:=(u3xu[i]-w3xu[i])/dd;

if xu[i]<=0 then

begin

writeln('Stężenie składnika o nr.',i,'w cieczy',

' wyczerpanej jest ujeane.');

writeln('NALEŻY SKORYGOWAC DANE W ZBIORZE DANE1');

halt;

end

end;

end;

end;

'c':begin

dd:=s(xu[i]-xu[i])/(xd[i]-xu[i]);

if dd <=0 then

begin

writeln('Natężenie przepływu destylatu jest ujeane');

```

        writeln('NALEZY SKORYGOWAC DANE W ZBIORZE DANE1');
        halt;
    end;
    w:=s-dd;
    if w <=0 then
    begin
        writeln('Natezenie cieczy wyczerpanej jest ujemne');
        writeln('NALEZY SKORYGOWAC DANE W ZBIORZE DANE1');
        halt;
    end;
end;
end; (case dana)
end; (if bil = 'n')
l:=r$dd;
v:=l+dd;
ll:=l+s;
kond: end; (koniec procedury bilans)

```

(zbiór FIZCHEM.pas)

```

procedure FIZCHEM;
label kon,pocz;
($i as\polki\podprogr\ulwagowe.pas)
($i as\polki\podprogr\wlfix.pas)
($i as\polki\podprogr\przelicz.pas)
begin
    writeln('');
    writeln('Czy dane sa natezenia przeplywu i srednie wlasciwosci');
    writeln(' fizykochemiczne strumieni w kolumnie ? (t/n)');
    readln(dan);
    if dan='n' then daki='wl.fix obliczane'
    else begin

```

```

      dsk:='wl.fiz sa dane';
      writeln('');writeln('');
      writeln('Program korzysta z wartosci ',
              ' zestawionychw zbiorze DANE');
      goto pocz;
    end;
assign(tabwar,'a:\polki\zbiordan\tabwar.pas');
reset(tabwar);
begin
  read(tabwar,es,nf,iw);
  for ii:=1 to es do read(tabwar,ms[ii]);
  for ii:=1 to es do read(tabwar,pch[ii]);
  for ii:=1 to es do read(tabwar,rch[ii]);
  for ii:=1 to es do read(tabwar,tkr[ii]);
  for ii:=1 to es do read(tabwar,prkr[ii]);
  for ii:=1 to es do
    for j:=1 to iw do read(tabwar,rc[ii,j]);
  for ii:=1 to es do
    for j:=1 to iw do read(tabwar,rg[ii,j]);
  for ii:=1 to iw do read(tabwar,ti[ii]);
  for ii:=1 to es do read(tabwar,aa[ii]);
  for ii:=1 to es do read(tabwar,ba[ii]);
  for ii:=1 to es do read(tabwar,ca[ii]);
  for ii:=1 to es do
    for j:=1 to ms do read(tabwar,ww[ii,j]);
  end;
  writeln('');
  writeln('Czy obliczana bedzie polka dla kolumny rektyfikacyjnej',
          ' ? (t/n)');
  readln(rekt);
  case rekt of
    't':begin
      writeln('');

```

```

writeln('Czy sporządzony masz bilans ogólny kolumny ?(t/n)');
readln(bila);

case bila of
'n':begin
    writeln(' ');
    writeln('Podaj wartość stopnia deflegacji R ');
    readln(r);writeln('');
    writeln('Brednie parametry fizykochemiczne cieczy ',
            'i par w kolumnie obliczane są');
    writeln(' przy założeniu, że temperatura w przekroju',
            ' zasilania ');
    writeln(' jest równa temperaturze wrzenia surowki !');
    writeln('                $$$$$$$$$$$$$$$$$$',
            '$$$$$$$$');
    writeln('WYBIERZ dla jakich ograniczeń procesowych ',
            'są być wykonane obliczenia');
    writeln('');
    writeln('a - narzucony jest skład i ilość destylatu ');
    writeln('b - zadany jest skład i ilość cieczy',
            ' wyczerpanej');
    writeln('c - znany jest skład destylatu i cieczy',
            ' wyczerpanej'); writeln('');
    writeln('WCISNIJ odpowiednio klawisz "a", "b" lub "c" ');
    readln(dana);writeln('');
    assign(dane1, 'a:\polki\zbior\dan\dane1.pas');
    reset(dane1);
    begin
        for j:=1 to nf do read(dane1, fs[j]);
        for j:=1 to nf do read(dane1, tf[j]);
        for j:=1 to nf do
            for ii:=1 to es do read(dane1, xf[ii, j]);
    case dana of
'a':begin

```



```

    read (dane1,dd);
    for i:=1 to ms do read(dane1,xd[i]);
end;
'b':begin
    read (dane1,w);
    for i:=1 to ms do read(dane1,xw[i]);
end;
'c':begin
    for i:=1 to ms do read(dane1,xd[i]);
    for i:=1 to ms do read(dane1,xw[i]);
end;
end;(case dane)
end;
end;(case bila='n')
't': begin
    writeln('');
    writeln('Program korzysta z wartosci zestawionych w',
            ' zbiorze DANE3');
    writeln('Do zbioru tego musza byc wprowadzone staze',
            ' nia skladnikow w surowce, cieczy wyczerpanej');
    writeln('i w destylacie (faza ciepla) oraz natezenia ',
            ' przeplywu surowki, cieczy wyczerpanej i dles',
            ' tylatu a takze temperatura surowki',
            ' i stopien deflegacji');
    assign (dane3,'a:\polki\zbiordan\dane3.pas');
    reset(dane3);
    for j:=1 to nf do
        for ii:=1 to ms do read(dane3,xff[ii,j]);
        for i:=1 to ms do read(dane3,xw[i]);
        for i:=1 to ms do read(dane3,xd[i]);
        for j:=1 to nf do read(dane3,fs[j]);
        for j:=1 to nf do read(dane3,tf[j]);
        read (dane3,w,dd,r);
    end;
end;

```

```

        end; (case bila= 't')
    end; (case bila of)
        writeln('Czy obliczana bedzie dolna czesc kolumny ? (t/n)');
        readln(reak);
        if reak='t' then rek:='ponizej zasilania'
            else rek:='powyzej zasilania';
        end; (case rekt = 't')
    'n':begin
        writeln('');
        writeln('Program korzysta z wartosci zestawionych ',
            'w zbiorze DANE2');
        writeln('');
        writeln('Do zbioru tego musza byc wprowadzone dla szczytu ',
            'i dolu kolumny');
        writeln('nastepujace wartosci:');
        writeln('1.Stezenia skladnikow w fazie cieklej i gazowej');
        writeln('2.Temperatury strumieni ');
        writeln('3.Natezenia przeplywu strumieni cieczy i par');
        assign(dane2,'a:\polki\zbiordan\dane2.pas');
        reset(dane2);
        begin
            for i:=1 to ms do read (dane2,xd[i]);
            for i:=1 to ms do read (dane2,yd[i]);
            for i:=1 to ms do read (dane2,xw[i]);
            for i:=1 to ms do read (dane2,yw[i]);
            read(dane2,tgc,tgg,tdc,tdg,lgc,vgg,ldc,vdg);
        end;
    'end; (case rekt='n')
end; (case rekt of)
    writeln('');
    writeln('Czy stezenia podane sa w ulamkach molowych?(t/n)');
    while KeyPressed do
        znak:=ReadKey;

```

```

readln(jed);
if jed='n' then
  case rekt of
    't':begin
      if bila='t' then dane:='c';
      ulwag;
    end;
    'n':begin
      przelicz(xw);
      przelicz(xd);
      przelicz(yw);
      przelicz(yd);
    end;
  end; (case rekt)
wlasfiz;
goto kon;
pocz:=assign(dane,'a:\polki\zbiordan\dane.pas');
reset(dane);
readln(dane,vg,vc,tp,roc,rog,etc,etg,ec,eg,si,alfa,vmol,axy);
kon:=end; (procedure FIZCHEM)

```

(zbiór 6E8T08C.pas)

```

function ror(x,ro:wektor;ac:real;al:wektor;ai:integer):real;
var i:integer;
    s:real;
begin
  s:=0;
  for i:=1 to m do
    s:=s+x[i]*al[i]/ro[i];
  ror:=ac/s;
end; (procedure ror)

```

(zbiór INTABLIC.pas)

```

procedure intab(t:real;ti:wektor;ri:tablica;iw,ni:integer;
               var ro:wektor);
  (interpolacja z dwuch tablic)
  var i1,i,k,n1,ni:integer;
      ma:real;
      ba:wektor;
  label et,ed;
  begin
    ni:=iw;
    repeat
      ma:=ti[i1];
      for i:=1 to m do
        ba[i]:=ri[i,1];
      ni:=1;
      for i1:=2 to n1 do
        if ma < ti[i1] then
          begin
            ma:=ti[i1];
            for i:=1 to m do
              ba[i]:=ri[i,i1];
            ni:=i1;
          end;
      ti[n1]:=ti[ni];
      for i:=1 to m do ri[i,ni]:=ri[i,n1];
      ti[n1]:=ma;
      for i:=1 to m do ri[i,n1]:=ba[i];
      ni:=ni-1;
    until ni=1;
    if t<ti[1] then
      begin
        ed: writeln('wartosc zmiennej poza przedzialem interpolacji t=',t);

```

```

writeln('NALEZY SKORYSOWAC DANE W ZBIORZE TABWAR');halt;
end;
for i:=1 to a do
begin
  if t=ti[i] then ro[i]:=ri[i,1] else
  begin
    ki:=1;
    etski:=k+1;
    if t <=ti[k] then
      ro[i]:=(t-ti[k-1])*(ri[i,k]-ri[i,k-1])/(ti[k]-ti[k-1])+ri[i,k-1]
      else if k<>lw then goto et else goto ed;
    end; (if t=ti[i])
  end; (for i)
end; (intab)

```

(zbior INTER2.PAB)

```

function indwa(a,b:real;a1,b1:wektor;c1:tabe;k,m:integer):real;
  (interpolacja funkcji dwu zmiennych)
  (%i a1\polki\podprogr\interpol.pas)
var j,lk:integer;
    c1:wektor;
    y1:wektor;
begin
  for j:=1 to a do
  begin
    for lk:=1 to k do y[lk]:=c1[j,lk];
    c[j]:=inter(a,y,a1,k);
  end; (for j)
  indwa:=inter(b,c,b1,m);
end; (procedure indwa)

```


(zbior INTERPOL.pas)

```

function inter(x:real;y,x1:wektor;n:integer):real;
($interpolacja funkcji jednej zmiennej$)
var k,i,m,n1:integer;
    aa,ba:real;
label cas;
begin
    n1:=n;
repeat
    aa:=x1[1]; ba:=y[1]; m:=1;
    for i:=2 to n1 do
        if aa < x1[i] then
            begin aa:=x1[i]; ba:=y[i]; m:=i end;
    x1[m]:=x1[n1]; y[m]:=y[n1];
    x1[n1]:=aa; y[n1]:=ba;
    n1:=n1-1;
until n1=1;
    k:=0; i:=1; init:=1;
    if (x<x1[1]) or (x>x1[n]) then
        begin
            k:=1;
            goto cas;
        end
    else if (x=x1[1]) then
        begin
            k:=2;
            goto cas;
        end
    else i:=2;
while (i<=n) and (k>4) do
begin
    if (x<=x1[i]) then k:=4
    else begin

```

```
i:=i+1;
```

```
end;
```

```
end;
```

```
case case k of
```

```
1:begin writeln('Wartosc x poza przedzialem interpolacji.',
```

```
      ' nastapi zatrzymanie programu');
```

```
      writeln('SKORYGUJ DANE');
```

```
      ini:=0
```

```
end;
```

```
2:inter:=y[i];
```

```
4:begin inter:=(x-x1[i-1])*(y[i]-y[i-1])/(x1[i]-x1[i-1])+y[i-1];
```

```
end;
```

```
end;(case k);
```

```
end;(Interpol)
```

(Zbiór KRYTERIA)

```
procedure KRYTERIA;
```

```
var k:integer;
```

```
begin
```

```
  clrscr;
```

```
  writeln('Przyjmij kryterium , wg.ktorego chcesz wybierac ',
```

```
    ' optymalna polke'); writeln('');
```

```
  writeln('Jesli nie odpowiada ci zadna z postaci 1 do 4 to mozesz ',
```

```
    ' ustalic własne');
```

```
  writeln('kryterium wprowadzajac odpowiednie zalany',
```

```
    ' w zbiorze KRYTWYB.');
```

```
  writeln('#####');
```

```
  writeln('§                                     §');
```

```
  writeln('§ 1. KRW = Kek / E                                     §');
```

```
  writeln('§                                     §');
```

```
  writeln('§ 2. KRW = Kek / E / Vp                                     §');
```

```

writeln('8
writeln('8 3. KRW = Ksk 8 pog / E
writeln('8
writeln('8 4. KRW = Ksk 8 pog / E / Vp
writeln('8
writeln('8 5. Po zatrzymaniu programu wywołaj 8');
writeln('8 zbior KRYTMYB i ustal własne 8');
writeln('8 kryterium.Mozesz przy tym ko - 8');
writeln('8 rzystac z dowolnych zmiennych 8');
writeln('8 występujących w programie 8');
writeln('8
writeln('#####');
writeln('');
writeln('WCISNIJ klawisz z numerem wybranej opcji');
readln(k);
case k of
1,2,3,4: kkr:=k;
5: halt;
end; (case k)
end; (procedury KRYTERIA)

```

(Zbiór KRYTMYB)

```

function kryt:real;
begin
case kkr of
1: kryt:=ksk/ek;
2: kryt:=ksk/ek/zp;
3: kryt:=ksk8pog/ek;
4: kryt:=ksk8pog/ek/zp;
end; (case kkr)
end; (function kryt)

```

(zbiór LEPCIEZ.pas)

```

function ter (t:real; x:wektor; m:integer; rch:wektor; rc, mc:real):real;
  (obliczanie lepkości mieszaniny cieczy Ns/m2)
(*1 a:\polki\podprogr\potega*)
var a, s:real;
    i:integer;
begin
  s:=0.0;
  for i:=1 to m do s:=s+x[i]*rch[i];
  a:=s*rc/mc;
  ter:=1e-28*pot(a,8);
end; (procedure ter)

```

(zbiór LEPGAZ.pas)

```

function teta (t:real; y:wektor; m:integer; ml, tkr, pkr:wektor):real;
(*1 a:\polki\podprogr\potega.pas*)
(obliczanie lepkości mieszaniny par Ns/m2)
var a, b, c, d, s, tr:real;
    i:integer;
    te:wektor;
begin
  b:=0; s:=0;
  for i:=1 to m do
    begin
      a:=pot(tkr[i], 0.1667)/(sqrt(ml[i])*pot(pkr[i], 0.6667));
      tr:=(t+273.15)/tkr[i];
      if tr<1.5 then te[i]:=34e-8*pot(tr, 0.94)/a
        else begin
          c:=4.58*tr-1.67;
          te[i]:=17.78e-8*pot(c, 0.625)/a;
        end
    end
  end;

```

```

                                end;

    d:=sqrt(m1[i]8tkr[i]);
    b:=b+y[i]8te[i]8d;
    a:=a+y[i]8d;
  end; (for i)
  teta:=b/a;
end; (procedure teta)

```

{zbior MABA.pas}

```

function mac(m1,x:wektor;m:integer):real;
{obliczanie addytywnych wlasciwosci fizykochemicznych
- masa czasteczkowa kg/kmol}
var i:integer;s:real;
begin
  s:=0.0;
  for i:=1 to m do
    s:=s+m1[i]8x[i];
  mac:=s;
end; (procedure mac)

```

{zbior NAPPON.pas}

```

function sir(x:wektor;ac,rc:real;pch:wektor;m:integer):real;
{obliczanie napiecia powierzchniowego mieszaniny cieczy}
{01 a:\polki\podprogr\potega.pas}
var p,s:real;
    i:integer;
begin
  s:=0;
  for i:=1 to m do

```



```

    w:=w+(1/3)*pch[4];
    p:=sqrt(w);
    sir:=1e-15*pot(p,4);
end; (procedure sir)

```

(zbiór POTEGA.pas)

```

function pot(a,b:real):real;
begin
    pot:=exp(b*ln(a))
end;

```

(zbiór PRZELEN.pas)

```

procedure przel(dk:real;var t,lp:real;var nlp,jlpi:integer);
(*ustalenie wartosci ze.pom. nlp przy zwiekszaniu lp, ewentualnie
zwiekszanie lp i t *)
var k,kl:integer;
begin
    case kl of
        1:case krok of
            1:begin
                t:=t+0.01;
                krok:=2;
                kl:=2;
            end;
        2:if jk=3 then
            begin
                t:=t+0.01;
                krok:=3;
                kl:=2;
            end;
    end;
end;

```

```

end
else k1:=1;
3: k1:=1;
end; (case krok przy k1=1)
2: case krok of
1: begin
t:=t+0.015; krok:=2; k1:=2
end;
2: begin
t:=t+0.01; krok:=3; k1:=2
end;
3: k1:=1;
end; (case krok przy k1=2)
3: case krok of
1: begin
t:=t+0.02; krok:=2; k1:=2
end;
2: begin
krok:=3; k1:=2
end;
3: k1:=1;
end; (case krok przy k1=3)
end; (case k1)
case k1 of
1: nlp:=1;
2: begin lp:=lp+0.18; nlp:=0; jlp:=jlp+1; end;
end; (case k1)
end; (procedure przelew)

```

{zbiór PRZELICZ.pas}

```

procedure PRZELICZ(var xx:wektor);
var i:integer;
    s:real;
begin
    s:=0;
    for i:=1 to ms do
        s:=s+xx[i]/m[i];
    for i:=1 to ms do
        xx[i]:=xx[i]/m[i]/s;
    end;
end;

```

{zbiór ROMKRO.pas}

```

procedure KRO(te,p:real;xe:tabel;j:integer;var k:wektor);
{ obliczanie współczynników równowagi fazowej jako funkcji
  temperatury, ciśnienia i składu }
var i,il,k:integer;
    a,b,c,pn,gam:real;
begin
    for k:=1 to ms do
        begin
            a:=0;b:=0;c:=0;
            for i:=1 to ms do
                begin
                    b:=b+xe[i,j]*mw[i,il];
                    c:=c+xe[i,j]*mw[i,k]/b;
                end;
            for il:=1 to ms do

```

```

as:=a+xe[i1,j]*ww[k,i1];
gam:=exp(1-ln(a)-c);
pn:=aa[k]-ba[k]/(te+ca[k]);
kr[k]:=gam*pot(10,pn)*133.323/pe;
end; (for k)
lsk:=kr[1]; nlsk:=1;
csk:=kr[1]; ncsk:=1;
for k:=2 to ms do
begin
  if lsk<kr[k] then begin
    lsk:=kr[k];
    nlsk:=k;
  end;
  if csk>kr[k] then begin
    csk:=kr[k];
    ncsk:=k;
  end;
end;
alfa:=lsk/csk;
end; (procedure KRO)

```

(zbior SPRAWNO.pas)

```

procedure sprawnosci;
{ Obliczenie sprawnosci kolumny z polkami przelewowymi oraz
powierzchni wymiany masy na tych polkach }
var kafi:integer;
    fip,u1,a2,u2,u3,f1,af,lcc,lq,a1,a3:real;
begin
  ns1:=288/0.35;
  ns1:=round(ns1);
  wa1:=vg/fa;

```

```

fip:=fo/fa;
u1:=wa$sqrt(rog);
a2:=-fip*fip$ai$(roc-rog);
u2:=4.424$pot(a2,0.25);
u3:=u1/u2;
f1:=pot(u3,0.28);
mig:=(1-f1)$sqr(u1/fip)/ai/2;
kgy:=7.6728f1$sqrt((roc-rog)/ai);
if u3>0.65 then kaf:=1 else
    if u3<=0.1 then kaf:=2 else
        kaf:=3;
case kaf of
    1:af:=mig;
    2:af:=kgy;
    3:af:=((mig-kgy)$u3-0.18mig+kgy)/0.55;
end; (case kaf)
lcc:=vc$roc/ac;
faf:=af$hp;
lg:=vg$rog/ag;
dyc:=7.4e-15$sqrt(as$ac)$ (tp+273.15)/etc/pot(vae1,0.6);
a1:=f/fo;a2:=lcc/lg;a3:=1000$hp;u2:=f/3600/vg/rog;
u1:=5.08$pot(a1,0.28)$pot(a2,0.024)$pot(a3,0.241)$pot(u2,0.013);
a1:=ai$f/etc/vg;a2:=etc/roc/dyc;
eag:=u1$pot(a1,0.044)$pot(a2,0.137)$pot(alfa,-0.028)$0.01;
des:=axy$lg/lcc;
emp:=eag/(1+eag$lg$ae$ag/ac/lcc);
ek:=ln(1+emp$(des-1))/ln(des);
end; ( procedure SPRAND8C )

```


(zbiór TEMPERAT.pas)

```

procedure TEMP(t0,pe:real;xe:stabel;j:integer;es:stabel;
               var tp:real;var yt:wekto.);
( Obliczanie temperatury wrzenia cieczy )
var k,i,j1,n:integer;
    tt,ft,eui:array[1..4] of real;
    r31,r32,r12,t1,t2,tas,tai,r1,r2,r3,r4,r5,r6,r7,r8,r9,a,b,c,
    del:real;
label et1,et4,et12,wdr1,et11,et7,et2,et10,koniec,et17,et18,et3,
    et6,et15,et16;
begin
    k:=1;
    et2:j1:=1;
    tt[j1]:=t0-58k;
    et1:su[j1]:=0;
    kro(tt[j1],pe,xe,j,kr);
    for i:=1 to es do
    begin
        su[j1]:=su[j1]+xe[i,j]skr[i]sas[i,j];
    end;
    ft[j1]:=su[j1]-1.0;
    if j1=4 then goto et4;
    if j1=3 then goto et12;
    j1:=j1+1;
    if j1=2 then tt[j1]:=t0 else tt[j1]:=t0+58k;
    goto et1;
et4: if abs(ft[4])<=0.001 then goto wdr1 else goto et11;
et12:if (ft[1]ft[2]<0.0) then
begin
    n:=1;
    goto et7;
end

```

```

else
    if (ft[2]*ft[3]<0.0) then
        begin
            n:=1; goto et7
        end
    else
        begin
            k:=k+1;
            goto et2
        end;

et7: if n<=20 then goto et10;
wdr1:tp:=tt[4];
    for i:=1 to m do
        ytf[i]:=xe[i,j]*kr[i]*es[i,j];
    goto koniec;
et10:r1:=tt[1]-tt[2];
    r2:=tt[2]-tt[3];
    r3:=tt[1]-tt[3];
    r31:=r3*r1;
    r12:=r1*r2;
    r32:=r3*r2;
    r4:=ft[1]/r31;
    r5:=ft[2]/r12;
    r6:=ft[3]/r32;
    r7:=tt[2]+tt[3];
    r8:=tt[1]+tt[3];
    r9:=tt[1]+tt[2]; a:=r4-r5+r6;
    b:=-r4*r7+r5*r8-r6*r9;
    c:=-4*r2*r3-r5*tt[1]*tt[3]+r6*tt[1]*tt[2];
    del:=b*b-4*a*c;
    if del<0.0 then
        begin
            writeln('nie ma pierwiastkow rzeczywistych',

```

```

(procedura TEMP)";

goto koniec
end;

del:=sqrt(del);
t1:=(-b+del)/2/a;
t2:=(-b-del)/2/a;

if tt[1]>tt[2] then tma:=tt[1]
               else tma:=tt[2];

if tma>tt[3] then goto et17 else tma:=tt[3];
et17:if tt[1]<tt[2] then tmi:=tt[1]
               else tmi:=tt[2];

if tmi<tt[3] then goto et18 else tmi:=tt[3];
et18:if t1<tma then
begin
  if t1>tmi then tt[4]:=t1 else tt[4]:=t2
  end else tt[4]:=t2;

j1:=4;
goto et1;

et11:if (ft[1]+ft[2]<0.0) then
begin
  if (ft[1]+ft[3]<0.0) then
begin
  if (abs(tt[4]-tt[2])<abs(tt[4]-tt[3])) then goto et6
  else goto et5
end else
  if (abs(tt[4]-tt[1])<abs(tt[4]-tt[3])) then goto et6
  else goto et15
end;

if (abs(tt[4]-tt[1])<abs(tt[4]-tt[2])) then goto et5
else goto et15;

et5: tt[2]:=tt[4];
ft[2]:=ft[4];goto et16;
et6: tt[3]:=tt[4];

```

```

    ft[3]:=ft[4];goto et14;
et15:tt[1]:=tt[4];
    ft[1]:=ft[4];
et16:n:=n+1;goto et7;
koniec:end; {procedure TEMP}

```

(zbiór ULWAGOME.pas)

```

procedure ULWAG;
(01 a:\polki\podprogr\przelicz.pas)
(01 a:\polki\podprogr\masa.pas)
var i,j :integer;
    sumak:real;
begin
    for j:=1 to nf do
        begin
            sumak:=0;
            for i:=1 to ms do
                xur[i]:=xf[i,j];
            przelicz(xur);
            for i:=1 to ms do
                sumak:=sumak+xur[i]*sm[i];
            fa[j]:=fa[j]/sumak;
            for i:=1 to ms do
                xf[i,j]:=xur[i];
        end; {for j}
    case dana of
        'a':begin
            przelicz(xd);dd:=dd/mac(ms,xd,ms)
        end;
        'b':begin
            przelicz(xw);w:=w/mac(ms,xw,ms)

```

```

    end;
    'c':begin
        przelicz(xw);
        przelicz(xd);
    end;
end; (case dana)
end; (procedure ULNAG)

```

{zbior MLASNOBC.pas}

```

procedure wlas(tg,tc:real;xp,yp:wektor);
var i:integer;
begin
    ac:=aac(am,xp,as);
    intab(tc,ti,rc,iw,as,ro);
    roc:=ror(xp,ro,ac,am,as);
    etc:=ter(tc,xp,as,rch,roc,ac);
    si:=sir(xp,ac,roc,pch,as);
    intab(tg,ti,rg,iw,as,ro);
    rog:=aac(ro,yp,as);
    etg:=teta(tg,yp,as,am,tkr,pkr);
    ag:=aac(am,yp,as);
end; (koniec procedury wlas)

```

{zbior MLFICH.pas}

```

procedure MLASFIZ;
var i,ii:integer;
    su:real;
{01 as\polki\podprogr\gestosc.pas}
{01 as\polki\podprogr\lspciecz.pas}

```



```

(01 a:\polki\podprogr\leppaz.pas)
(01 a:\polki\podprogr\masa.pas)
(01 a:\polki\podprogr\nappow.pas)
(01 a:\polki\podprogr\bilansow.pas)
(01 a:\polki\podprogr\rowkro.pas)
(01 a:\polki\podprogr\temperat.pas)
(01 a:\polki\podprogr\wlasnosc.pas)
begin
  case rekt of
    't': begin
      bilans;
      for i:=1 to ms do
        for ii:=1 to nf do
          eofi,ii:=1.0;
      su:=0;
      for j:=1 to nf do
        begin
          temp(tfi[j],pcal,xf,j,eo,tsu[j],yp);
          su:=su+tsu[j];
        end;
      ts:=su/nf;tg:=ts;tc:=ts;
      wlas(tg,tc,xs,yp);
      mcs:=mc;ags:=ag;
      rocs:=roc;rogs:=rog;
      etcs:=etc;etgs:=etg;
      nls:=sl;alfas:=alfa;
      xpri:=xs[nlsk]/(xs[nlsk]+xs[ncek]);
      mxy:=alfa/(1+(alfa-1)*xpri);
      case reak of
        't':begin
          t0:=ts+5;
          for iii:=1 to 4 do
            begin

```

```

      xpf[i1]:=xw[i1];
      xwr[i1,1]:=xw[i1];
    end;
    temp(t0,pcal,xwr,1,so,tp,yp);
    xpri:=xw[nlsl]/(xw[nlsl]+xw[nclsl]);
    mxy:=(mxy+alfa/(1+(alfa-1)*xpri))/2;
    i:=i1;
  end; (case reak = 't')
'n':begin
  t0:=ts-5;
  for i:=1 to ms do
  begin
    xpf[i1]:=xd[i1];
    xwr[i1,1]:=xd[i1];
  end;
  temp(t0,pcal,xwr,1,so,tp,yp);
  xpri:=xd[nlsl]/(xd[nlsl]+xd[nclsl]);
  mxy:=(mxy+alfa/(1+(alfa-1)*xpri))/2;
  end; (case reak = 'n')
end; (case reak)
tg:=tp;tc:=tp;
end; (case rekt = 't')
'n': begin
  tg:=tgg;tc:=tgc;
  for i:=1 to ms do
    for i1:=1 to nf do
      xf[i,i1]:=xd[i1];
    krotg,pcal,xf,1,kr);
    xpri:=xd[nlsl]/(xd[nlsl]+xd[nclsl]);
    mxy:=alfa/(1+(alfa-1)*xpri);
    for i1:=1 to ms do
      begin
        xxf[i1]:=xd[i1]; ypf[i1]:=yd[i1];

```

```

end;

vs:=(vvg+vdg)/2; ls:=(lqc+ldc)/2;
wlas(tg,tc,xs,yp);
acs:=ac;ags:=ag;
rocs:=roc;rogs:=rog;
etc:=etc;etgs:=etg;
sis:=si;alfas:=alfa;
tcs:=tdc;tgs:=tdg;
for i:=1 to ms do
  for ii:=1 to l do
    xf[i,ii]:=xw[i];
    kro(tg,pcal,xf,i,kr);
    xpri:=(xw[lnsk]/(xw[lnsk]+xw[ncsk]));
    xmy:=(mxy+alfa/(1+(alfa-1)*xpri))/2;
    for l:=1 to ms do
      begin
        xpfil:=xw[i]; ypfil:=yw[i];
      end
    end; (case rekt = 'n')
end; (case rekt of)
wlas(tg,tc,xp,yp);
acs:=(ac+acs)/2;
ags:=(ag+ags)/2;
rocs:=(roc+rocs)/2;
rogs:=(rog+rogs)/2;
etc:=(etc+etc)/2;
etgs:=(etg+etgs)/2;
sis:=(si+sis)/2;
alfas:=sqrt(alfas*alfa);
vg:=vsg/rog/3600;
vc:=lsc/roc/3600;
vso:=sg/rog;
end; (procedure MLBF12)

```

{xbior WYBBRED.pas}

```

procedure srednica(ds:vector;il:integer;var jk:integer);
var j:integer;
label ser;
begin
  for j:=1 to il do
    if d<= ds[j] then
      begin
        d:=ds[j];jk:=j;goto ser;
      end
    else if j=il then
      begin
        writeln('srednica wieksza od',
          ' znormalizowanej');
        jk:=0;
      end;
  ser:;end;{ procedure BREDNICA }

```

10. PRZYKŁAD OBLICZEŃ

Przykładowe obliczenia przeprowadzono dla półki kolpakowej przeznaczonych do kolumny rektyfikacyjnej. Korzystano z danych zestawionych w zbiorze DAMEI wybierając opcję "c". Obliczenia przeprowadzono dla układu alkohol etylowy - kwas octowy - octan etylu - woda. Dane fizykochemiczne składników zestawiono w zbiorze TABMAR. Za podstawę wyboru optymalnego rozwiązania przyjęto kryterium K1 (równanie 4).

10.1. Dane

(Zbiór DANKOLP)

101325.0 13 10 7 2.6

(Zbiór TABKOLP)

288.9	461.1	572.2	633.3	683.3	700.0				
0.30	0.4	0.5	0.6	0.7	0.75				
0.6	0.8	1.0	1.1	1.2	1.4				
0.25	0.4	0.55	0.7	0.88	0.98	1.1	1.2	1.35	
1.50	1.60	1.80	1.90	2.00					
2.5	5.0	7.5	10.0	15.0	20.0	30.0	40.0	50.0	
60.0	70.0	80.0	90.0	100.0					
0.2	0.4	0.6	1.0	1.5	2.0	3.0	4.0	6.0	
8.0	10.0	15.0	20.0	30.0					
5.0	10.0	20.0	30.0	40.0	50.0	64.0			
0.5	0.6	0.7	0.8						
0.53	0.58	0.65	0.71	0.79	0.91	0.94			
0.69	0.72	0.79	0.83	0.87	0.94	0.97			
0.86	0.88	0.91	0.92	0.94	0.97	0.98			

(Zbiór DANE1)

100 91

0.226 0.3557 0.209 0.2093
 0.0001 0.4301 0.0402 0.5297
 0.4218 0.3239 0.1823 0.072

(Zbiór TABWAR)

4 1 8

60.0 46.0 18.0 88.0
 133.5 127.5 54.2 217.1
 75.3 76.1 21.0 118.4
 594.75 516.25 677.30 523.25
 57.20 62.96 218.40 37.99
 1017.5 1006.0 994.8 983.5 971.8 959.9 948.3 936.2
 763.3 754.1 743.6 734.8 725.1 715.7 705.7 692.5
 988.1 983.2 977.8 971.8 965.3 958.4 951.0 943.5
 863.6 850.8 837.6 824.5 811.2 797.2 783.1 768.3
 0.3100 0.4621 0.6730 0.9590 1.3380 1.8330 2.4680 3.2710
 0.5060 0.7900 1.1900 1.7400 2.5000 3.5100 4.8600 6.5600
 0.0831 0.1302 0.1982 0.2934 0.4235 0.5977 0.8264 1.1210
 1.2000 1.8000 2.5610 3.4950 4.6770 6.1580 8.0060 10.300
 50.0 60.0 70.0 80.0 90.0 100.0 110.0 120.0
 7.5596 7.83124 7.96681 7.09808
 1644.05 1440.52 1668.21 1238.71
 233.524 212.71 228.0 217.0
 1.0 0.27558 0.26838 0.6179
 2.2818 1.0 0.15347 0.7667
 1.22642 0.92038 1.0 0.14907
 0.89277 0.55046 0.12333 1.0

10.2. Wyniki obliczeń

Poniżej zestawiono wydruki wyników ze zbiorów WYKOLP i WYBKOLP.
(Zbiór WYKOLP)

WYNIKI OBLICZEN POLEK KOLPAKOWYCH

dla założonych parametrów ruchomych (V_g, V_c, R)

$V_g=2.4074 \text{ m}^3/\text{s}$ $V_c=0.00467 \text{ m}^3/\text{s}$ $R=5.0$
powyżej zasilania

*** $H=0.30$ $hp=0.00$ *** kolpaki nie wykorzystane
*** $H=0.35$ $hp=0.00$ *** kolpaki nie wykorzystane
*** $H=0.40$ $hp=0.052$ *** za małe pokrycie wyciek przy przepływie
cieczy

```

#####
*** H=0.40 m      D=2.00 m      F=3.1416 m2      wg=0.7661 m/s
                    KP0=395.21      KP=298.42      KSK=693.64
** hp=0.057 m      lp=1.500m      Ek=0.4839      KRW=1433.39
         t           dk           fa           fo           hs
         [m]           [m]           [m2]           [m2]           [m]
         0.150          0.100          2.4376          0.3317          0.0144
         S            bp            fp            hpe            hpp
         [m]           [m]           [m2]           [m]           [m]
         0.6614          1.6427          0.3520          0.1531          0.0445
         ik            ir            roz            ns            hr
         [szt]          [szt]          [-]           [-]           [m]
         97            9            a            4            0.0700
         hwr           hpk           hwp           hwc           hwl
         [m]           [m]           [m]           [m]           [m]
         0.0210          0.0140          0.0080          0.0262          0.0301
         pr            pg            pt            hg            pog
         [N/m2]          [N/m2]          [N/m2]          [m]           [N/m2]

```

6.56	379.83	137.26	0.0076	591.14
ta	me	hpl	hc	zp
[m]	[kg/kg]	[m]	[m]	[m3]
0.0528	0.0058	0.1514	0.0391	0.1362

=====

*** H=0.40 hp=0.062 *** za duzy poziom cieczy w elemencie splywowym

*** H=0.45 hp=0.052 *** za male pokrycie wyciec przy przeplywie
cieczy

=====

*** H=0.45 m	D=2.00 m	F=3.1416 m2	wg=0.7661 m/s	
	KPO=395.21	KP=335.73	KBK=730.94	
** hp=0.057 m	lp=1.500m	Ek=0.4844	KRW=1509.07	
t	dk	fa	fo	hs
[m]	[m]	[m2]	[m2]	[m]
0.150	0.100	2.4376	0.3317	0.0144
B	bp	fp	hpe	hpp
[m]	[m]	[m2]	[m]	[m]
0.6614	1.8427	0.3520	0.1531	0.0445
ik	ir	roz	ns	hr
[szt]	[szt]	[--]	[--]	[m]
97	9	a	4	0.0700
hwr	hpk	hwp	hwc	hw1
[m]	[m]	[m]	[m]	[m]
0.0210	0.0140	0.0080	0.0262	0.0301
pr	pq	pt	hg	pog
[N/m2]	[N/m2]	[N/m2]	[m]	[N/m2]
6.56	379.83	137.26	0.0076	591.14
ta	me	hpl	hc	zp
[m]	[kg/kg]	[m]	[m]	[m3]
0.0570	0.0038	0.1514	0.0391	0.1362

=====

*** H=0.45 hp=0.062 *** za duzy poziom cieczy w elemencie splywowym

*** H=0.50 m D=1.80 m F=2.5447 m2 wg=0.9459 m/s

KPD=323.06 KP=341.15 KSK=664.22

** hp=0.052 m lp=1.170 m Ek=0.4364 KRW=1522.14

t	dk	fa	fo	hs
[m]	[m]	[m2]	[m2]	[m]
0.140	0.100	2.1986	0.3625	0.0172
B	bp	fp	hpe	hpp
[m]	[m]	[m2]	[m]	[m]
0.6839	1.6073	0.1730	0.1541	0.0395
ik	ir	roz	ns	hr
[szt]	[szt]	[--]	[--]	[m]
106	10	a	4	0.0700
hwr	hpk	hwp	hwc	hw1
[m]	[m]	[m]	[m]	[m]
0.0230	0.0120	0.0050	0.0283	0.0283
pr	pg	pt	hg	pog
[N/m2]	[N/m2]	[N/m2]	[m]	[N/m2]
13.69	343.58	114.94	0.0123	572.27
ta	me	hpi	hc	zp
[m]	[kg/kg]	[m]	[m]	[m3]
0.0662	0.0036	0.1613	0.0370	0.0946

*** H=0.50 hp=0.057 *** za duzy poziom cieczy w elemencie splywowym

*** H=0.55 m D=1.80 m F=2.5447 m2 wg=0.9459 m/s

KPD=323.06 KP=375.27 KSK=698.33


```

** hp=0.052 m      lp=1.170 m      Ek=0.4366      KRW=1599.66
  t      dk      fa      fo      hs
  [m]      [m]      [m2]      [m2]      [m]
0.140      0.100      2.1986      0.3625      0.0172
  S      bp      fp      hpe      hpp
  [m]      [m]      [m2]      [m]      [m]
0.6839      1.6073      0.1730      0.1541      0.0395
  ik      ir      roz      ns      hr
  [mzt]      [mzt]      [-]      [-]      [m]
106      10      a      4      0.0700
  hwr      hpk      hwp      hwc      hw1
  [m]      [m]      [m]      [m]      [m]
0.0230      0.0120      0.0050      0.0283      0.0283
  pr      pg      pt      hg      pog
  [N/m2]      [N/m2]      [N/m2]      [m]      [N/m2]
13.69      343.58      114.94      0.0123      572.27
  ta      me      hpl      hc      zp
  [m]      [k6/k6]      [m]      [m]      [m3]
0.0702      0.0026      0.1613      0.0370      0.0946

```

*** H=0.55 hp=0.057 *** za duzy poziom cieczy w elemencie splywowym

```

*** H=0.60 m      D=1.80 m      F=2.5447 m2      wg=0.9459 m/s
      KP0=323.06      KP=409.38      KBK=732.45
** hp=0.052 m      lp=1.170 m      Ek=0.4367      KRW=1677.36

```

```

  t      dk      fa      fo      hs
  [m]      [m]      [m2]      [m2]      [m]
0.140      0.100      2.1986      0.3625      0.0172
  S      bp      fp      hpe      hpp
  [m]      [m]      [m2]      [m]      [m]

```

0.6839	1.6073	0.1730	0.1541	0.0395
ik	ir	roz	ns	hr
[szt]	[szt]	[--]	[--]	[m]
106	10	a	4	0.0700
hwr	hpk	hwp	hwc	hw1
[m]	[m]	[m]	[m]	[m]
0.0230	0.0120	0.0050	0.0283	0.0283
pr	pg	pt	hg	pog
[N/m2]	[N/m2]	[N/m2]	[m]	[N/m2]
13.69	343.58	114.94	0.0123	572.27
ta	me	hpi	hc	xp
[m]	[kg/kg]	[m]	[m]	[m3]
0.0741	0.0020	0.1613	0.0370	0.0946

#####

*** H=0.60 hp=0.057 *** za duzy poziom ciaczy w elemencie splywowym

#####

*** H=0.65 m D=1.80 m F=2.5447 m2 wg=0.9459 m/s

KP0=323.06 KP=443.50 KBK=766.56

** hp=0.052 m lp=1.170 m Ek=0.4367 KRW=1755.16

t	dk	fa	fo	hs
[m]	[m]	[m2]	[m2]	[m]
0.140	0.100	2.1986	0.3625	0.0172
B	bp	fp	hpe	hpp
[m]	[m]	[m2]	[m]	[m]
0.6839	1.6073	0.1730	0.1541	0.0395
ik	ir	roz	ns	hr
[szt]	[szt]	[--]	[--]	[m]
106	10	a	4	0.0700
hwr	hpk	hwp	hwc	hw1
[m]	[m]	[m]	[m]	[m]

0.0230	0.0120	0.0050	0.0283	0.0283
pr	pg	pt	hg	pog
[N/m2]	[N/m2]	[N/m2]	[m]	[N/m2]
13.69	343.58	114.94	0.0123	572.27
ta	me	hpl	hc	zp
[m]	[k6/k6]	[m]	[m]	[m3]
0.0777	0.0016	0.1613	0.0370	0.0946

*** H=0.65 hp=0.057 *** za duzy poziom cieczy w elemencie splywowym

*** H=0.70 m	D=1.80 m	F=2.5447 m2	wg=0.9459 m/s
	KP0=323.06	KP=477.62	KBK=800.68
** hp=0.052 m	lp=1.170 m	Ek=0.4368	KRW=1833.03

t	dk	fa	fo	hs
[m]	[m]	[m2]	[m2]	[m]
0.140	0.100	2.1986	0.3625	0.0172
B	bp	fp	hpe	hpp
[m]	[m]	[m2]	[m]	[m]
0.6839	1.6073	0.1730	0.1541	0.0395
ik	ir	roz	ns	hr
[szt]	[szt]	[~]	[~]	[m]
106	10	a	4	0.0700
hwr	hpk	hwp	hwc	hwl
[m]	[m]	[m]	[m]	[m]
0.0230	0.0120	0.0050	0.0283	0.0283
pr	pg	pt	hg	pog
[N/m2]	[N/m2]	[N/m2]	[m]	[N/m2]
13.69	343.58	114.94	0.0123	572.27
ta	me	hpl	hc	zp
[m]	[k6/k6]	[m]	[m]	[m3]

0.0812

0.0012

0.1613

0.0370

0.0946

#####

*** H=0.70 hp=0.057 *** za duzy poziom cieczy w elemencie splynowym

*** H=0.75 hp=0.052 *** polka niestabilna

*** H=0.75 hp=0.057 *** polka niestabilna

*** H=0.75 hp=0.062 *** nierowny przeplyw cieczy

*** H=0.75 hp=0.067 *** nierowny przeplyw cieczy

*** H=0.75 hp=0.072 *** nierowny przeplyw cieczy

*** H=0.75 hp=0.077 *** nierowny przeplyw cieczy

*** H=0.75 hp=0.082 *** nierowny przeplyw cieczy

(Zbiór WYBKOLP)

WYNIKI OBLICZEN POLEK KOLPAKOWYCH

dla zalozonych parametrow ruchowych (Vg, Vc, R)

Vg=2.4074 m3/s

Vc=0.00467 m3/s

R= 5.0

powyzej zasilania

*** H=0.40 m

D=2.00 m

F=3.1416 m2

wg=0.7661 m/s

KP0=395.21

KP=298.42

KEK=693.64

** hp=0.057 m

lp=1.500m

Ek=0.4839

KRW=1433.39

t

dk

fa

fo

hs

[m]

[m]

[m2]

[m2]

[m]

0.150

0.100

2.4376

0.3317

0.0144

8

bp

fp

hpe

hpp

[m]

[m]

[m2]

[m]

[m]

0.6614

1.8427

0.3520

0.1531

0.0445

ik

ir

roz

ns

hr

[szt]

[szt]

[--]

[--]

[m]

97

9

a

4

0.0700

hwr	hpk	hwp	hwc	hw1
[m]	[m]	[m]	[m]	[m]
0.0210	0.0140	0.0080	0.0262	0.0301
pr	pg	pt	hg	pog
[N/m2]	[N/m2]	[N/m2]	[m]	[N/m2]
6.56	379.83	137.26	0.0076	591.14
ta	me	hpi	hc	xp
[m]	[kG/kG]	[m]	[m]	[m3]
0.0528	0.0058	0.1514	0.0391	0.1362

11 WYKAZ OZNACZEN

Schematy	Programy		Jednostki
aa	aa	tablica współczynników a w równaniu Antoine'a [30] do obliczania prężności pary nasyconej składników	
as	as	współczynnik uwzględniający stopień asocjacji	
a _{max}	am	maksymalna wysokość podniesienia zaworka	m
a _{min}	ami	minimalna wysokość podniesienia zaworka	m
b	b	stosunek l/D_p	
ba	ba	tablica współczynników b w równaniu Antoine'a [30] do obliczania prężności pary nasyconej składników	
bila	bila	zmienna pomocnicza = 'n', gdy ma być obliczany bilans kolumny, w przeciwnym przypadku = 't'	
b _{max}	bma	maksymalne podniesienie kołpaka	m
b _{min}	bi	minimalne podniesienie kołpaka	m
b _p	bp	średnia szerokość półki	m
b1	b1	tablica l/D_p przy obliczaniu E [19]; ZAWOR	
b2	b2	tablica l/D_p przy obliczaniu E [34]; KOLP, 8ITO	
b3	b3	tablica współczynników przy obliczaniu poprawki Δ [8]; KOLP	
C	c	stała przy obliczaniu dozwolonej prędkości gazu wg Boundersa i Browna [34]	

ca	ca	tablica współczynników c w równaniu Antoine'a [30] do obliczania prężności pary nasyconej składników	
Cg	cg	współczynnik przy obliczaniu Δ_j KOLP	
ch	ch	spiętrzenie cieczy na półce sicowej, przy założeniu, że nie płynie gaz	a
chi	chi	minimalna wartość ch	a
chm	chm	maksymalna wartość ch	a
chl	chl	tablica wartości argumentu funkcji do obliczania współczynnika k_j BITO	
Cs	cs	współczynnik korygujący Δ [34]; KOLP	
csi	csi	tablica współczynników Cs [34]; KOLP	
C _v	kk	stała zależna od f_p/f_r przy obliczaniu Δp_{cz}	
ci	ci	tablica wartości funkcji do obliczania współczynnika w rów. Boundersa-Browna [34]; KOLP, BITO	
c2	c2	tablica wartości funkcji współczynnika cg [34]; KOLP	
c3	c3	tablica współczynników przy obliczaniu poprawki Δ [8]; KOLP	
dan	dan	zmienna pomocnicza = 't', gdy własności fizykochemiczne są znane, w przeciwnym przypadku = 'n'	
d _k	dk	średnica zewnętrzna kołpaka; KOLP	a
d _o	dot(do)	średnica otworka; BITO, ZAWOR, BITBEZ	a
doi	doi	tablica d _o ; BITO tablica d _{sz} ; RUSZT	
d _r	dr	średnica rurki gazowej; KOLP	a
d _{sz}	dot(do)	szerokość szczeliny; RUSZT	a
d _z	dz	średnica zewnętrzna zaworka; ZAWOR	a
di	di	tablica wartości D [29]	

d2	d2	tablica wartości D półki dwustrumieniowej [28]; ZAWOR	
D	d	średnica kolumny	m
Des	des	współczynnik desorpcji	
D _{or}	dor	średnica orientacyjna	m
e	ae	porywanie cieczy przez fazę gazową	kg/kg
e1	e1	tablica wartości E [34]; KOLP, BITO, ZAWOR	
E	e	współczynnik przewężenia przy obli- czaniu h_g [34]; KOLP, BITO, ZAWOR	
E _g	eg	sprawność lokalna półki	
E _k	ek	sprawność kolumny	
E _{Hg}	eah	sprawność umowna półki	
fgr	fgr	tablica wartości poprawki przy obliczaniu Δ [8]; KOLP	
fi1	fi1	tablica wartości ϕ ; BITBEZ, RUBZT	
faf	faf	powierzchnia międzyfazowa	m ²
f _{max}	fa	powierzchnia dla przepływu gazu przy a_{max} ; ZAWOR	m ²
f _{min}	fai	powierzchnia dla przepływu gazu przy a_{min} ; ZAWOR	m ²
f _p	fp	przekrój pierścienia między rurką gazową a kołpakiem; KOLP	m ²
fpb	fpb	powierzchnia przelewu bocznego na półce dwustrumieniowej; ZAWOR	m ²
fps	fps	powierzchnia przelewu środkowego na półce dwustrumieniowej; ZAWOR	m ²
f _r	fr	przekrój rurki gazowej; KOLP	m ²
		przekrój szczeliny między przelewem, a półką; BITO	m ²
f _a	fa	tablica natężeń przepływu strumieni surówki	
f _w	fw	sumaryczna powierzchnia wycięć	

		Jednego kołpaka; KOLP	m ²
f_v	f_v	współczynnik obciążenia [20]; ZAWOR	
f_z	f_z	sumaryczna powierzchnia zaworków na półce; ZAWOR	m ²
F	f	powierzchnia przekroju kolumny	m ²
F_a	f_a	powierzchnia aktywna półki	m ²
F_g	f_g	powierzchnia przekroju półki bez wpływu; BITO	m ²
F_o	f_o	sumaryczna powierzchnia otworków; BITO; ZAWOR; BITBEZ	m ²
F_p	f_p	powierzchnia przelewu	m
F_{sz}	$f_o(F_{sz})$	sumaryczna powierzchnia szczeliny; RUBZT	m ²
F_w	f_w	powierzchnia roboczej części półki (pokrytej otworkami); BITO	m ²
g_a	g_a	tablica grubości zaworka [26]; ZAWOR	
g_{z1}	g_{z1}	grubość zaworków lżejszych; ZAWOR	m
g_{z2}	g_{z2}	grubość zaworków cięższych; ZAWOR	m
h_c	h_c	wysokość cieczy na półce	m
h_k	h_k	wysokość kołpaka; KOLP	m
h_{k1}	h_{k1}	tablica wartości h_{pk}	
h_p	h_p	wysokość krawędzi przelewowej	m
h_{p1}	h_{p1}	tablica wartości h_p	
h_{pe}	h_{pe}	wysokość cieczy w elemencie spływowym mierzona od dna półki	m
h_{pk}	h_{pk}	odległość kołpaka od dna półki	m
h_{pp}	h_{pp}	odległość między dolną krawędzią przelewu, a półką	m
h_r	h_r	długość rurki kominka; KOLP	m
		wysokość zredukowana cieczy; BITBEZ, RUBZT	m
h_{rn}	h_{rn}	długość rurki kominka wg normy [25]; KOLP	m

h_s	hs	spiętrzenie cieczy na przelewie	m
h_{smax}	hsa	maksymalna wartość h_s ; SITO	m
h_{smin}	hsl	minimalna wartość h_s ; SITO	m
h_{sp}	hsp	odległość dolnej krawędzi elementu spływowego od górnej krawędzi prze- lewu odpływowego	m
h_w	hw	wysokość wycięć w kołpakach KOLP	m
h_{wc}	hwc	odległość górnej krawędzi szczelin kołpaka od średniego poziomu cieczy na półce; KOLP	m
h_{wp}	hwp	odległość górnej krawędzi wycięć kołpaka od górnej krawędzi przelewu	m
h_{wr}	hwr	odległość górnej krawędzi wycięć kołpaka od górnej krawędzi rurki gazowej	m
h'_w	hwl	wysokość wycięć kołpaka zajęta przez gaz	m
h_z	hz	wysokość zredukowana cieczy; KOLP	m
$h1$	h1	tablica wartości H [29]	
$h2$	h2	tablica argumentów funkcji, której wartości wczytano do tablicy c1; KOLP, SITO	
H	h	odległość między półkami	m
H_e	he	odległość czoła piany od dna półki	m
H_p	hpl	wysokość piany na półce	m
ib	ib	wymiar tablicy l_{p1} ; SITO	
id	id	wymiar tablicy d_1 ; ZAMOR	
ido	ido	wymiar tablic do_1 ; sg ; RUSZT; BITBEZ	
ifi	ifi	wymiar tablicy fi_1 ; RUSZT; BITBEZ	
ih	ih	wymiar tablicy h_1	
ik	ik	ilość kołpaków na półce; KOLP	
ip	ip	wymiar tablicy d_2 ; ZAMOR	
ir	ir	ilość rzędów kołpaków na półce	

iw	iw	wymiar tablic t_1 , r_c , r_g	
jd	jd	wymiar tablicy d_1 ; KOLP, SITO	
k	k	wymiar tablicy h_{p1} ; SITO, ZAWOR	
k_{p1}	kp	koszt płaszcza między dwiema półkami	j.pieniężne
k_{p01}	kpo	koszt półki	j.pieniężne
ks	ks	współczynnik przy obliczaniu A_p [15]; SITO	
kw	kw	współczynnik we wzorze Eduljee [7] do obliczania w_{min} ; SITO	
k1	k1	tablica wartości współczynników kw	
k2	k2	tablica wartości współczynników ks	
KRW	krw	kryterium wyboru	
l_c	lc	długość drogi cieczy na półce	m
l_p	lp	długość krawędzi przelewowej	m
$lp1$	lp1	tablica wartości l_p/D ; SITO	
L	l	natężenie przepływu cieczy; ZAWOR zmienna pomocnicza; SITO	m^3/h
L_d	lgc	natężenie przepływu cieczy w górnym przekroju kolumny	kmol/h (kg/h)
L_w	ldc	natężenie przepływu cieczy w dolnym przekroju kolumny	kmol/h (kg/h)
m	m	wymiar tablicy h_{p1} ; KOLP tablica współczynników równania do obliczania f_v [20]	
ma	ma	tablica wartości m_z [26]; ZAWOR	
ma	ma	tablica mas cząsteczkowych składników	
ms	ms	liczba składników w układzie	
m_{xy}	mxy	nachylenie linii równowagowej	
mz	mz	średnia masa zaworków na półce	kg
mz1	mz1	masa zaworka lżejszego	kg
mz2	mz2	masa zaworka cięższego	kg
mi	mi	tablica współczynników równania do obliczania f_v [20]	

M_c	mc	średnia masa cząsteczkowa cieczy	kg/kmol
M_g	mg	średnia masa cząsteczkowa fazy gazowej	kg/kmol
na	na	maksymalna wartość stosunku $n1/n2$	
nb	nb	tablica wartości S_p, l_p, F_p, F_a [24]; ZAWOR	
nbl	nbl	tablica wartości $sp1, sp2, lp11, lp2, fpb, fps, F_a$ [28]; ZAWOR	
ncsk	ncsk	numer składnika ciężkiego kluczowego	
nf	nf	liczba strumieni zasilaających kolumnę	
nlsk	nlsk	numer składnika lekkiego kluczowego	
norfo	norfo	tablica wartości F_o [23]; BITO, BITBEZ	
norir	norir	tablica wartości ir [22]; KOLP	
norma	norma	tablica wartości ik [22]; KOLP	
norroz	norroz	tablica zmiennej roz [22]; KOLP	
ns	ns	liczba stref pełnego wyaleszania cieczy na półce	
n_w	nw	liczba wycięd w kołpaku [25]; KOLP	
n_z	nz	średnia liczba zaworków na półce; ZAWOR	
n1	n1	liczba zaworków lżejszych; ZAWOR	
n2	n2	liczba zaworków cięższych; ZAWOR	
N_{og}	nog	liczba jednostek przenikania masy	
Δp_c	pc	opór warstwy cieczy	N/m^2
Δp_{cz}	pt	częściowa strata ciśnienia na półce; KOLP	N/m^2
Δp_g	pg	opór fazy gazowej	N/m^2
Δp_s	ps	minimalny opór półki suchej; BITO	N/m^2
Δp_{og}	pog	całkowita strata ciśnienia na półce	N/m^2
Δp_r	pr	strata ciśnienia spowodowana odrywaniem się pęcherzyków gazowych	N/m^2
Δp_s	ps	opór półki suchej	N/m^2
Δp_{sa}	psa	opór półki suchej przy minimalnym	

		przepływie gazu	N/m^2
Pch	pch	tablica wartości parachory składników	
Pkr	pkp	tablica wartości ciśnień krytycznych składników	
reak	reak	zmienna pomocnicza = 't', gdy obliczana jest dolna część kolumny oraz = 'n' w przypadku obliczania części górnej	
rekt	rekt	zmienna pomocnicza = 't', gdy obliczenia dotyczą kolumny rektyfikacyjnej	
rc	rc	tablica gęstości składników fazy ciekłej jako funkcja temperatury	
rg	rg	tablica gęstości składników fazy gazowej jako funkcja temperatury	
roz	roz	sposób rozmieszczenia kołpaków na półce [22]; KOLP	
R	r	stopień deflegacji	
R_v	rv	współczynnik rozdziału gazu na półce	
Rch	rch	tablica wartości reochory składników	
S	s	odległość przelewu od środka półki	m
sl	sl	tablica grubości półki; SITO	
smi	smi	tablica argumentów funkcji, której wartości zapisano w tabl.y1; BITBEZ, RUSZT	
S_p	sp	strzałka przelewu	m^2
s2	s2	tablica argumentów funkcji, której wartości zapisano w tabl.k2; SITO	
t	t	podziałka kołpaków; KOLP	m
		podziałka zaworków; ZAWOR	m
ta	ta	szerokość spływu zajęta przez ciecz	m
t_c	tc	temperatura cieczy	$^{\circ}C$
t_{cd}	tcg	temperatura cieczy na szczycie kolumny	$^{\circ}C$
t_{cw}	tdc	temperatura cieczy na dole kolumny	$^{\circ}C$

t_{gd}	tg	temperatura gazu na szczycie kolumny	$^{\circ}\text{C}$
t_{gw}	tdg	temperatura gazu na dole kolumny	$^{\circ}\text{C}$
t_g	tg	temperatura fazy gazowej	$^{\circ}\text{C}$
tf	tf	tablica temperatur strumieni surowki	
ti	ti	tablica temperatur, w których dane są wartości w tablicach rc i rg	
t_o	tot(to)	podziałka otworów na półce sitowej, bezprzelewowej; SITBEZ	m
t_{sz}	tot(tsz)	podziałka szczelin na półce rusztowej; RUSZT	m
t_w	tw	podziałka wycięd w kołpaku [25]; KOLP	m
tz	tz	typ zaworki; ZAWOR	
Tkr	tkr	tablica temperatur krytycznych składników	
V	v	natężenie przepływu fazy gazowej; ZAWOR	m^3/h
V_c	vc	średnie natężenie przepływu cieczy	m^3/s
V_g	vg	średnie natężenie przepływu fazy gazowej	m^3/s
V_d	vdg	natężenie przepływu fazy gazowej w górnym przekroju kolumny	kmol/h (kg/h)
V_p	zp	objętość cieczy zatrzymanej na półce	m^3
V_w	vvg	natężenie przepływu fazy gazowej w dolnym przekroju kolumny	kmol/h (kg/h)
V_{mol}	vmol	objętość molowa fazy gazowej	m^3/kmol
V_o	vo	zmienna pomocnicza; ZAWOR	
vi	vi	tablica argumentów funkcji, której wartości zapisano w tablicy c2	
w	w	prędkość dozwolona gazu odniesiona do pełnego przekroju kolumny	m/s
	wg	prędkość gazu w otworkach; ZAWOR	m/s
ww	ww	tablica współczynników Wilsona [33]	
w_a	wa	prędkość gazu odniesiona do F_a	m/s

w_c	w_c	prędkość cieczy w elemencie spływowym	m/s
w_g	w_g	prędkość gazu odniesiona do F	m/s
w_{min}	w_i	minimalna prędkość gazu w kolumnie z półkami bezprzelewowymi; SITBEZ, RUSZT	m/s
	w_{ei}	minimalna prędkość gazu potrzebna do utrzymania cieczy na półce; SITO	m/s
w_{max}	w_m	maksymalna prędkość gazu w kolumnie z półkami bezprzelewowymi; SITBEZ, RUSZT	m/s
w_o	w_o	prędkość gazu w otwórkach; SITBEZ, SITO	m/s
w_r	w_r	prędkość cieczy w przekroju między dolną krawędzią przelewu, a półką	m/s
x_d	x_d	tablica stężeń składników w destylacie lub w cieczy na szczycie kolumny	kmol/kmol (kg/kg)
x_f	x_f	tablica stężeń składników w zasileniu	kmol/kmol (kg/kg)
x_{pri}	x_{pri}	pseudoudział składnika lekkiego kluczowego w fazie ciekłej	
x_w	x_w	tablica stężeń w cieczy wyczerpanej	kmol/kmol (kg/kg)
x_l	x_l	tablica argumentów funkcji, której wartości zapisano w tablicy csl	
y_d	y_d	tablica stężeń w fazie gazowej w górnym przekroju kolumny	kmol/kmol (kg/kg)
y_w	y_w	tablica stężeń w fazie gazowej w dolnym przekroju kolumny	kmol/kmol (kg/kg)
y_l	y_l	tablica wartości funkcji do obliczenia współczynnika przy ustalaniu w_{min} [1]; SITBEZ, RUSZT	
y_2	y_2	tablica argumentów funkcji, której wartości zapisano w tablicy el	

α	alfa	współczynnik lotności względnej składnika lekkiego kluczowego	
β	bet	zmienna pomocnicza; RUBZT, SITBEZ	
Δ	hg	spiętrzenie cieczy odgazowanej na półce; KOLP	m
η_c	etc	dynamiczny współczynnik lepkości cieczy	Ns/m ²
η_g	etg	dynamiczny współczynnik lepkości gazu	Ns/m ²
ξ	ksi	współczynnik oporów; RUBZT, SITBEZ	
ρ_c	roc	gęstość cieczy	kg/m ³
ρ_g	rog	gęstość fazy gazowej	kg/m ³
ρ_p	rop	względna gęstość piany	kg/m ³
σ	si	napięcie powierzchniowe	N/m
τ	tau	współczynnik oporów; RUBZT, SITBEZ	
ϕ	fi	stosunek t_o/d_o ; SITO, SITBEZ	
		stosunek t_{sz}/d_{sz} ; RUBZT	
ϕ_f	ff	współczynnik obciążenia zalewającego [20]; ZAWOR	
ϕ_p	wp	współczynnik poprawkowy na spienienie [20]; ZAWOR	



12. WYKAZ LITERATURY

1. Aleksandrov J.A., Rektifikacjonnye i absorbcionnye apparaty. Chimijskaja. Moskwa 1971.
2. Aleksandrov J.A., Skoblo A.I., Mechaničeskij unos židkosti gazom s tarel'ok proval'nogo tipa. Chis. i tehn. Topliv., 1960, 5, 42.
3. Anonim., For your cost data file..., Brit. Chem. Eng., 1967, 12, 747.
4. Bandrowski J., Troniewski L., Destylacja i rektyfikacja. PWN, Warszawa 1980.
5. Bretsznajder St., Własności gazów i cieczy. PNT, Warszawa 1962.
6. Eduljee H.E., The design of grid-type trays., Brit. Chem. Eng., 1959, 4, 320.
7. Eduljee H.E., The design of sieve-type distillation plates. Brit. Chem. Eng., 1959, 4, 320.
8. Eduljee H.E., Design of bubble-cap distillation plates. Brit. Chem. Eng., 1964, 9, 820.
9. Engineering Staff Shell Development Company Emeryville, California, Turbogrid Distillation Trays. Chem. Eng. Progress, 1954, 50, 57.
10. English G.E., van Winkle M., Efficiency of fractionating columns. Chem. Eng., 1963, 70, 241.
11. Foldes P., Evangelidi I., Efficiency of turbogrid-tray distillation columns. Brit. Chem. Eng., 1968, 13, 1291.
12. Hoppel J., Ekonomia procesów przemysłu chemicznego. PNT, Warszawa, 1962.
13. Hobler T., Czajka J., Hydraulika półki sitowej i rusztowej. Chem. Stos., 1959, 8, 13.
14. Hobler T., Narzeczka L., Podsumowanie własnych badań hydrauliki półek rusztowych i sitowych z badaniami innych autorów. Chem. Stos.

- 1975, 18, 45.
13. Hoppe K., Grundlagen der Dimensionierung von kolonnenboden. Verlag Teodor Steinkopff. Drezno, 1967.
 14. Huel M., Standart G., The efficiency of large turbogrid-trays. Brit. Chem. Eng., 1966, 11, 1370.
 15. Hunt C., Hanson D.N., Wilke C.R., Capacity factors in the performance of perforated-plate columns. AIChE Journal, 1955, 1, 441.
 16. Kaštanek F., Efficiencies of different types of distillation plates. Collection Czechoslov. Chem. Commun., 1970, 35, 1170.
 17. Koziol A., Koch R., Metodyka przeliczania kolumn półkowych - półki kołpakowe na zaworkowe. Prace Naukowe Inst. Inż. Chem. i U. C. Politechniki Wrocławskiej. Wrocław 1972.
 18. Kuciel E., Projektowanie kolumn półkowych do wymiany masy za pomocą elektronowych maszyn cyfrowych. Prace Naukowe Inst. Inż. Chem. i U.C. Politechniki Wrocławskiej Nr 17, Monografie Nr 9. Wrocław 1973.
 19. PN-76/C-01350. Procesy podstawowe inżynierii chemicznej. Nazwy i ich symbole.
 20. BN-71/2231-01. Aparaty typu kolumnowego. Półki kołpakowe jedno-przelewowe. Podstawowe wielkości.
 21. BN-71/2231-02. Aparaty typu kolumnowego. Półki sitowe. Podstawowe wielkości.
 22. BN-73/2231-03. Aparaty typu kolumnowego. Półki zaworkowe jedno-przelewowe.
 23. BN-76/2232-01. Aparaty typu kolumnowego. Kołpaki okrągłe stalowe.
 24. BN-85/2232-03. Aparaty kolumnowe. Wewnętrzne wyposażenie kolumn. Zaworki okrągłe.
 25. BN-75/2232-06. Aparaty typu kolumnowego. Przegrody przelewowe.
 26. BN-85/2232-07. Aparaty kolumnowe. Półki zaworkowe dwuprzelewowe. Podstawowe wielkości.
 27. BN-85/2201-05. Aparaty kolumnowe. Średnice i odstęp między półkami.
 28. Praca zbiorowa. Poradnik fizykochemiczny. MNT. Warszawa 1974.

31. Stichlmair J., Punktverstärkungsverhältnis bei der Rektifikation und Absorption. VDI-Berichte, 1976, 260, 267.
32. Šušteršič L. J., Aerov M. E., Bystrova T. A., O gidrodinamičeskoe razčete kolonn s besperevalivnymi taratkami. Chim. Prom., 1962, 7, 530.
33. Suzuki I., Komatsu H., Hirata M., Formulation and prediction of quaternary vapour - liquid equilibria accompanied by esterification. J. Chem. Eng. Japan, 1970, 3, 152.2.
34. Ziolkowski Z., Destylacja i rektyfikacja w przemyśle chemicznym. WNT. Warszawa 1978.



Opisoby Rohumnowe - projektowanie
Inżynieria chemiczna
Programy specjalne

Biblioteka Główna
Politechniki Szczecińskiej

CHII.6573



004-006573-00-0

PI 959

PI 959