

Podstawy informatyki I r.

Zadania z programowania w języku Pascal.

Znajdujące się na poniższej liście zadania przeznaczone są do samodzielnego rozwiązania w czasie poza regularnymi zajęciami z przedmiotu podstawy informatyki.

Zadanie uważamy za rozwiązane poprawnie jeśli:

- a. program kompiluje się poprawnie do kody wykonywalnego, na dowolnym kompilatorze języka Pascal (należy unikać stosowania instrukcji, poleceń i funkcji charakterystycznych dla konkretnych implementacji),
- b. program wykonywalny działa zgodnie z określoną w zadaniu specyfikacją, w przeciwnym przypadku, na początku pliku źródłowego (przed słowem kluczowym **program**), w komentarzu musi zostać zawarta dokładna informacja na temat poczynionych odstępstw od specyfikacji (np. uproszczeń, założeń).

Rozwiązanie każdego zadania musi być umieszczone w osobnym pliku o rozszerzeniu ***.pas**. Nazwa pliku musi mieć postać: **zadanie_nr.pas** np. plik z rozwiązaniem zadania trzeciego nazywa się **zadanie_3.pas**. Rozwiązane zestaw zadań przed przesłaniem musi zostać umieszczony w katalogu: **nazwisko_studenta/ np. nowak/**. Tak przygotowany katalog wraz z zawartością należy spakować zipem: **zip lr nazwisko_studenta.zip nazwisko_studenta/**.

Przykład: **zip lr nowak.zip nowak**

Spakowane archiwum należy przesłać, w postaci załącznika do emaila, do dnia stycznia 10 stycznia 2002 na adres <mailto:topa@wszib.edu.pl>. W temacie listu należy umieścić "Zadania Pascal".

Literatura:

1. N. Wirth, *Algorytmy + struktury danych = programy*
2. M. Iglewski, J. Madey, S. Matwin, *Pascal*
3. N. Wirth, *Wstęp do programowania systematycznego*

Życzymy powodzenia!

Krzysztof Boryczko, Bartosz Baliś, Paweł Topa, Dariusz Żbik

1. Napisz program w języku Pascal, który wczytuje ciąg liczb rzeczywistych zakończony 0 (zerem). Zero kończące ciąg **nie** jest elementem ciągu, a jedynie znakiem specjalnym używanym do zaznaczenia końca ciągu. Wczytanie kolejnego elementu ma być poprzedzone znakiem zachęty, zawierającym prośbę o podanie elementu o konkretnym numerze (np.: Podaj element nr 2 > _). Po wczytaniu, program ma podawać długość najdłuższego rosnącego podciągu we wczytanym ciągu, oraz pozycje pierwszego i ostatniego elementu tego podciągu. Program powinien prawidłowo reagować na przypadek, gdy we wczytanym ciągu nie ma żadnego podciągu rosnącego o długości co najmniej 2.

Przykład:

Podaj element 1 > 1
Podaj element 2 > 4
Podaj element 3 > 7
Podaj element 4 > 1
Podaj element 5 > 3

Podaj element 6 > 3
Podaj element 7 > 3
Podaj element 8 > 0
Długość najdłuższego rosnącego podciągu: 3
Pozycja pierwszego elementu: 1
Pozycja ostatniego elementu: 3

2. Napisz program w języku Pascal, który wczytuje ciąg liczb rzeczywistych zakończony 0 (zerem). Zero kończące ciąg **nie** jest elementem ciągu, a jedynie znakiem specjalnym używanym do zaznaczenia końca ciągu. Wczytanie kolejnego elementu ma być poprzedzone znakiem zachęty, zawierającym prośbę o podanie elementu o konkretnym numerze (np.: Podaj element nr 2 > _). Po wczytaniu, program ma podawać długość najdłuższego podciągu we wczytanym ciągu składającego się z takich samych elementów oraz pozycje pierwszego i ostatniego elementu tego podciągu, a także wartość powtarzającego się elementu.

Przykład:

Podaj element 1 > 1
Podaj element 2 > 1
Podaj element 3 > 7
Podaj element 4 > 1
Podaj element 5 > 19
Podaj element 6 > 19
Podaj element 7 > 19
Podaj element 8 > 0
Długość najdłuższego podciągu takich samych elementów: 3
Powtarzający się element: 19
Pozycja pierwszego elementu: 5
Pozycja ostatniego elementu: 7

3. Napisz program w języku Pascal, który wczytuje ciąg liczb rzeczywistych zakończony 0 (zerem). Zero kończące ciąg **nie** jest elementem ciągu, a jedynie znakiem specjalnym używanym do zaznaczenia końca ciągu. Wczytanie kolejnego elementu ma być poprzedzone znakiem zachęty, zawierającym prośbę o podanie elementu o konkretnym numerze (np.: Podaj element nr 2 > _). Po wczytaniu, program ma podawać sumę elementów najdłuższego malejącego podciągu we wczytanym ciągu, oraz pozycje pierwszego i ostatniego elementu tego podciągu. Program powinien prawidłowo reagować na przypadek, gdy we wczytanym ciągu nie ma żadnego malejącego podciągu o długości co najmniej 2.

Przykład:

Podaj element 1 > 1
Podaj element 2 > 4
Podaj element 3 > 3
Podaj element 4 > 1
Podaj element 5 > 7
Podaj element 6 > 3
Podaj element 7 > 6
Podaj element 8 > 0
Suma elementów najdłuższego malejącego podciągu: 8
Pozycja pierwszego elementu: 2
Pozycja ostatniego elementu: 4

4. Napisz program w języku Pascal, który wczytuje ciąg liczb całkowitych zakończony 0 (zerem). Zero kończące ciąg **nie** jest elementem ciągu, a jedynie znakiem specjalnym używanym do zaznaczenia końca ciągu. Wczytanie kolejnego elementu ma być poprzedzone znakiem zachęty, zawierającym prośbę o podanie elementu o konkretnym numerze (np.: Podaj element nr 2 > _). Po wczytaniu, program ma podawać wartość maksymalnego parzystego i maksymalnego nieparzystego elementu ciągu oraz pozycje, na których te elementy wystąpiły. Program powinien prawidłowo reagować na przypadki, gdy we wczytanym ciągu nie ma żadnego elementu parzystego bądź nieparzystego.

Przykład:

Podaj element 1 > 1

Podaj element 2 > 15
Podaj element 3 > 3
Podaj element 4 > 9
Podaj element 5 > 7
Podaj element 6 > 4
Podaj element 7 > 6
Podaj element 8 > 0
Maksymalny element parzysty: 6, pozycja: 7
Maksymalny element nieparzysty: 15, pozycja: 2

5. Napisać program, który wczytuje ciąg liczb całkowitych zakończony zerem i wypisuje ilość liczb parzystych inieparzystych wtym ciągu. Wczytywane liczby mógł być ujemne. Przy wprowadzaniu danych program powinien pytać o każdą liczbę wraz z podaniem jej kolejnego numeru.

Przykład:

Podaj ciąg liczb:
Liczba 1 --> 2
Liczba 2 --> 3
Liczba 3 --> 2
Liczba 4 --> 7
Liczba 5 --> 0
Ilość liczb parzystych: 2
Ilość liczb nieparzystych: 2
Czy powtorzyc (t/n)? n

6. Napisać program, który wczytuje ciąg liczb rzeczywistych zakończony zerem, a następnie wypisuje dwie największe liczby w tym ciągu i podaje pozycje na których wystąpiły. Następnie program powinien zapytać, czy powtórzyć cykl i w razie odpowiedzi twierdzącej ('t') powtarzać cały proces, zaś w przypadku odpowiedzi negatywnej ('n') konczyć działanie. Wczytywane liczby mógł być ujemne. Przy wprowadzaniu danych program powinien pytać o każdą liczbę wraz z podaniem jej kolejnego numeru.

Przykład:

Podaj ciąg liczb:
Liczba 1 --> 2.3
Liczba 2 --> 32.7
Liczba 3 --> -122
Liczba 4 --> 77.7
Liczba 5 --> 0
Najwieksza liczba: 77.7, pozycja: 4
Druga najwieksza liczba: 32.7, pozycja: 2
Czy powtorzyc (t/n)? n

7. Napisać program, który wczytuje ciąg liczb całkowitych zakończony zerem, a następnie wypisuje sumę średnią arytmetyczną parzystych wyrazów tego ciągu. Dalej program powinien zapytać, czy powtórzyć cykl i w razie odpowiedzi twierdzącej ('t') powtarzać cały proces, zaś w przypadku odpowiedzi negatywnej ('n') konczyć działanie. Wczytywane liczby mógł być ujemne. Przy wprowadzaniu danych program powinien pytać o każdą liczbę wraz z podaniem jej kolejnego numeru.

Przykład:

Podaj ciąg liczb:
Liczba 1 --> 2
Liczba 2 --> 3
Liczba 3 --> 5
Liczba 7 --> 0
Suma: 23
Czy powtorzyc (t/n)? n

8. Napisać program, który wczytuje ciąg liczb całkowitych zakończony zerem, a następnie wypisuje sumę parzystych oraz sumę nieparzystych wyrazów tego ciągu. Następnie program powinien

zapytać, czy powtórzyć cykl i w razie odpowiedzi twierdzącej ('t') powtarzać cały proces, zaś w przypadku odpowiedzi negatywnej ('n') kończyć działanie. Wczytywane liczby mogą być ujemne. Przy wprowadzaniu danych program powinien pytać o każdą liczbę wraz z podaniem jej kolejnego numeru. W wypadku, gdy ciąg nie zawiera wyrazów parzystych lub nieparzystych, program powinien wyświetlać informację o braku takich elementów.

Przykład:

Podaj ciąg liczb:

Liczba 1 --> 2

Liczba 2 --> -2

Liczba 3 --> 33

Liczba 4 --> 1

Liczba 5 --> 0

Suma wyrazów parzystych: 0

Suma wyrazów nieparzystych: 32

Czy powtórzyć (t/n)? t

Podaj ciąg liczb:

Liczba 1 --> 2

Liczba 2 --> 4

Liczba 3 --> 6

Liczba 4 --> 0

Suma wyrazów parzystych: 12

Brak elementów nieparzystych.

Czy powtórzyć (t/n)? n

9. Napisać program, który wczytuje ciąg liczb całkowitych zakończony zerem, a następnie wypisuje ilość i średnią arytmetyczną parzystych wyrazów tego ciągu. Jeśli ciąg nie zawiera parzystych wyrazów, to program powinien wypisywać informację "Brak parzystych wyrazów". Następnie program powinien zapytać, czy powtórzyć cykl i w razie odpowiedzi twierdzącej ('t') powtarzać cały proces, zaś w przypadku odpowiedzi negatywnej ('n') kończyć działanie. Wczytywane liczby mogą być ujemne. Przy wprowadzaniu danych program powinien pytać o każdą liczbę wraz z podaniem jej kolejnego numeru.

Przykład:

Podaj ciąg liczb:

Liczba 1 --> 2

Liczba 2 --> 2

Liczba 3 --> 9

Liczba 4 --> 4

Liczba 5 --> 0

Ilość parzystych wyrazów: 3

Średnia arytmetyczna: 2.67

Czy powtórzyć (t/n)? t

Podaj ciąg liczb:

Liczba 1 --> 3

Liczba 2 --> 5

Liczba 3 --> 7

Liczba 4 --> 0

Brak parzystych wyrazów.

Czy powtórzyć (t/n)? n

10. Napisać program, który wczytuje ciąg liczb całkowitych zakończony zerem, a następnie wypisuje sumę parzystych oraz sumę nieparzystych wyrazów tego ciągu. Następnie program powinien zapytać, czy powtórzyć cykl i w razie odpowiedzi twierdzącej ('t') powtarzać cały proces, zaś w przypadku odpowiedzi negatywnej ('n') kończyć działanie. Wczytywane liczby mogą być ujemne. Przy wprowadzaniu danych program powinien pytać o każdą liczbę wraz z podaniem jej kolejnego numeru. W wypadku, gdy ciąg nie zawiera wyrazów parzystych lub nieparzystych, program powinien wyświetlać informację o braku takich elementów.

Przykład:

Podaj ciąg liczb:

Liczba 1 > 2

Liczba 2 --> -2

Liczba 3 --> 33

Liczba 4 --> -1

Liczba 5 --> 0

Suma wyrazów parzystych: 0

Suma wyrazów nieparzystych: 32

Czy powtórzyć (t/n)? t

Podaj ciąg liczb:

Liczba 1 --> 2

Liczba 2 --> 4

Liczba 3 --> 6

Liczba 4 --> 0

Suma wyrazów parzystych: 12

Brak elementów nieparzystych.

Czy powtórzyć (t/n)? n

11. Dana jest następująca definicja stałych i typów:

const

rozmiar = 100;

type

tablica = array[1..rozmiar] of integer;

Napisz program w języku Pascal, który do zdefiniowanej zmiennej typu *tablica* wczytuje ciąg liczb całkowitych zakończony 0 (zerem) jednak nie dłuższy niż wartość stałej *rozmiar* elementów. W dalszej kolejności we wczytanej tablicy program znajduje liczby parzyste i kopiuje je do drugiej zdefiniowanej w programie tablicy. W programie powinny znaleźć się trzy podprogramy (procedury lub funkcje) służące do **1**) wczytania liczb do pierwszej tablicy, **2**) wyszukiwania w pierwszej tablicy liczb parzystych oraz kopiowania ich do drugiej tablicy (podprogram powinien zwracać informacje o ilości liczb w drugiej tablicy), **3**) wypisywania elementów tablicy.

Przykład:

Tablica po wczytaniu:

7 -5 4 8 90

Tablica liczb parzystych:

4 8 90

Pamiętaj o wcięciach i komentarzach w kodzie programu.

12. Dana jest następująca definicja stałych i typów:

const

rozmiar = 100;

type

tablica = array[1..rozmiar] of integer;

Napisz program w języku Pascal, który do zdefiniowanej zmiennej typu *tablica* wczytuje ciąg liczb całkowitych zakończony 0 (zerem) jednak nie dłuższy niż wartość stałej *rozmiar* elementów. W dalszej kolejności we wczytanej tablicy program zmienia miejscami element największy oraz najmniejszy. Program wypisuje elementy tablicy wczytanej i zmodyfikowanej. W programie powinny znaleźć się cztery podprogramy (procedury lub funkcje) służące do **1**) wczytania liczb do tablicy, **2**) wyszukiwania pozycji elementów największego i najmniejszego, **3**) zamiany

elementów największego i najmniejszego w tablicy, **4)** wypisywania elementów tablicy.

Przykład:

Tablica po wczytaniu:

12 -5 4 8 90

Tablica po modyfikacji:

12 90 4 8 -5

13. Dana jest następująca definicja stałych i typów:

```
const
    rozmiar = 100;
type
    tablica = array[1..rozmiar] of real;
```

Napisz program w języku Pascal, który do zdefiniowanej zmiennej typu *tablica* wczytuje ciąg liczb całkowitych zakończony 0 (zerem) jednak nie dłuższy niż wartość stałej *rozmiar* elementów. Dodatkowo program wczytuje liczbę-wzorzec typu **real**. Program w tych wszystkich miejscach tablicy, gdzie znajdują się liczby o wartości większej od wartości podanej liczby-wzorca wpisuje liczbę 0 (zero). Program wypisuje liczby po wczytaniu ich do tablicy, wartość wzorca oraz elementy tablicy po modyfikacji. W programie powinny znaleźć się trzy podprogramy (procedury lub funkcje) służące do **1)** wczytania liczb do tablicy oraz wczytania wzorca, **2)** zerowania odpowiednich elementów tablicy, **3)** wypisywania elementów tablicy.

Przykład:

Tablica po wczytaniu:

-4.0 12.0 6.0 8.0 19.0

Wzorzec: 7,

Tablica po modyfikacji:

-4.0 0.0 6.0 0.0 0.0

14. Dana jest następująca definicja stałych i typów:

```
const
    rozmiar = 100;
type
    tablica = array[1..rozmiar] of integer;
```

Napisz program w języku Pascal, który do zdefiniowanej zmiennej typu *tablica* wczytuje ciąg liczb całkowitych zakończony 0 (zerem) jednak nie dłuższy niż wartość stałej *rozmiar* elementów. Następnie program zmienia kolejność elementów w tablicy na odwrotną (tzn. zamienia pierwszy z ostatnim, drugi z przedostatnim, itd.). Program wypisuje elementy tablicy wczytanej i zmodyfikowanej. W programie powinny znaleźć się trzy podprogramy (procedury lub funkcje) służące do **1)** wczytania liczb do tablicy, **2)** zmiany kolejności elementów w tablicach, **3)** wypisywania elementów tablicy.

Przykład:

Tablica po wczytaniu:

-3 17 4 8 90

Tablica po modyfikacji:
90 8 4 17 -3

15. Dana jest następująca definicja stałych i typów:

```
const
    rozmiar = 100;
type
    tablica = array[1..rozmiar] of real;
```

Napisz program w języku Pascal, który do zdefiniowanej zmiennej typu *tablica* wczytuje ciąg liczb całkowitych zakończony 0 (zerem) jednak nie dłuższy niż wartość stałej *rozmiar* elementów. Dla wczytanych liczb program oblicza ich średnią arytmetyczną, a następnie w tych miejscach tablicy, gdzie znajdują się liczby o wartości mniejszej od obliczonej średniej arytmetycznej wpisuje liczbe 0 (zero). Program wypisuje liczby po wczytaniu ich do tablicy, średnią arytmetyczną oraz elementy tablicy po modyfikacji. W programie powinny znaleźć się trzy podprogramy (procedury lub funkcje) służące do **1)** wczytania liczb do tablicy, **2)** obliczania średniej arytmetycznej i zerowania odpowiednich elementów tablicy, **3)** wypisywania elementów tablicy.

Przykład:

Tablica po wczytaniu:
-3.0 2.0 4.0 8.0 90.0

Średnia arytmetyczna: 20.20,

Tablica po modyfikacji:
0.0 0.0 0.0 0.0 90.0

16. Dana jest następująca definicja stałych i typów:

```
const
    rozmiar = 100;
type
    tablica = array[1..rozmiar] of integer;
```

Napisz program w języku Pascal, który do zdefiniowanej zmiennej typu *tablica* wczytuje ciąg liczb całkowitych zakończony 0 (zerem) jednak nie dłuższy niż wartość stałej *rozmiar* elementów. Dla podanych liczb program znajduje medianę (medianą jest liczba, która w posortowanej tablicy zawierającej nieparzystą liczbę elementów zajmuje środkowe miejsce, a jeżeli liczba elementów jest parzysta, to medianą jest średnia arytmetyczna z dwóch liczb środkowych). W programie powinny znaleźć się trzy podprogramy (procedury lub funkcje) służące do **1)** wczytania liczb do tablicy, **2)** posortowania liczb w tablicy, **3)** wyznaczenia i wypisania mediany.

Przykład:

Dla ciągu:
-3 2 4 8 90 mediana to liczba 4,

zaś dla ciągu:
-5 7 9 12 56 78 mediana wynosi: $(9+12)/2 = 10.5$.

17. Z klawiatury wprowadzany jest ciąg liczb całkowitych zakończonych zerem. Zero nie należy do ciągu. Wczytywany ciąg umieszczany jest w tablicy jednowymiarowej. Program najpierw sortuje wczytany ciąg w porządku malejącym. Z klawiatury wczytywana jest następnie liczbę całkowitą wzorzec, dla której program stara się znaleźć jej pozycję jej pierwszego wystąpienia w ciągu. Jeśli

podany wzorzec nie występuje w ciągu określana jest pozycja liczby o wartości najbliższej do podanego wzorca. Wczytany i posortowany ciąg oraz wyniki poszukiwania wzorca mają być wypisane na ekran. W programie należy zdefiniować funkcje i procedury:

1. funkcja wczytująca ciąg liczb z klawiatury. Funkcja zwraca ilość wczytanych liczb.
2. procedurę sortującą wczytany ciąg dowolną metodą w porządku malejącym.
3. funkcję, która jako jeden z argumentów przyjmuje wczytaną liczbę - wzorzec i zwraca jej pozycję w ciągu.
4. procedura drukująca ciąg zawarty w tablicy na ekranie.

Przykład:

ciąg: 3, 5, 7, 1, 8, 34, 8, 3, 21, 16
posortowany: 1, 3, 3, 5, 7, 8, 8, 16, 21, 34,
Podaj liczbę: 8
Pozycja w ciągu: 6
Podaj liczbę: 14
Liczba 14 nie ma w ciągu. Pozycja liczby najbliższej: 8

18. Z klawiatury wprowadzany jest ciąg liczb rzeczywistych zakończonych zerem. Zero nie należy do ciągu; jak zwykle pełni rolę markera końca. Wczytywany ciąg umieszczany jest w tablicy jednowymiarowej. Najpierw program sortuje ciąg liczb. Następnie z klawiatury wczytywana jest liczba całkowita - wzorzec - dla której znajdowany jest w ciągu podzbiór liczb większych od niej lub równych. Znaleziony zbiór liczb w postaci posortowanej zapisywany jest do drugiej tablicy. Funkcja zwraca długość znalezionego ciągu. Obydwa ciągi muszą być wydrukowane na ekranie. W programie należy zdefiniować funkcje i procedury:

1. funkcja wczytująca ciąg liczb z klawiatury. Funkcja zwraca ilość wczytanych liczb.
2. procedura sortująca wczytany ciąg dowolną metodą w porządku malejącym.
3. funkcja, która jako jeden z argumentów przyjmuje wczytaną liczbę - wzorzec i wyznacza ciąg liczb większych od niej lub równych wg przedstawionej powyżej specyfikacji.
4. procedurę, która drukuje ciąg zawarty w tablicy na ekran, liczby mają być podawane z dokładnością do 2 miejsca po przecinku.

Przykład:

ciąg: 3.6, 5.96, 7.1, 1.77, 8, 34.7, 8.34, 3.5, 21, 16.6
posortowany: 1.77, 3.5, 3.6, 5.96, 7.1, 8, 8.34, 16.6, 21, 34.7
Podaj liczbę: 6
Ciąg liczb większych od 6 zawiera 6 liczb: 7.1, 8, 8.34, 16.6, 21, 34.7

19. Z klawiatury wprowadzany jest ciąg liczb całkowitych zakończonych zerem. Zero nie należy do ciągu; jak zwykle pełni rolę markera końca. Wczytywany ciąg umieszczany jest w tablicy jednowymiarowej. Program z wczytanego ciągu wybiera liczby parzyste i zapisuje je w drugiej tablicy. Następnie obie tablice są sortowane w porządku rosnącym. W programie należy zdefiniować i użyć następujących funkcji i procedur:

1. funkcja wczytująca ciąg z klawiatury. Funkcja zwraca ilość wczytanych liczb.
2. funkcję wyznaczającą zbiór liczb parzystych wczytanego ciągu. Funkcja zwraca ilość znalezionych liczb parzystych.
3. procedura sortująca ciąg w tablicy dowolną metodą.
4. procedura, która drukuje ciąg zawarty w tablicy na ekran.

Przykład:

ciąg: 3, 5, 7, 1, 8, 34, 8, 3, 21, 16
 liczby parzyste: 8, 8, 34, 16
 posortowany ciąg: 1, 3, 3, 5, 7, 8, 8, 16, 21, 34,
 posortowany ciąg liczb parzystych: 8, 8, 16, 34

20. Pewna funkcja matematyczna f , określona na zbiorze liczb całkowitych, dana jest w postaci tabeli wartości tej funkcji dla różnych x np:

x : 4, 6, 12, 8, 5, 9, 8, 2,
 $f(x)$: 16, 18, 34, 24, 21, 24, 6

Wartości x oraz odpowiadające im wartości $f(x)$ wprowadzamy z klawiatury. Wprowadzanie kończy się z chwilą gdy jako x podamy zero. Powtórzenie wartości x podczas wprowadzania należy zasygnalizować jako błąd i powtórzyć wprowadzanie liczby.

Wartości x i odpowiadające im $f(x)$ mają być przechowywane w tablicy rekordów. Rekord składa się z dwóch pól, z których

pierwsze przechowuje x , drugie $f(x)$. Program sortuje tablicę wg pola zawierającego wartość x .

Następnie, z klawiatury

wczytywana jest liczba całkowita - wzorzec - dla której wyznaczana jest odpowiadająca jej wartość $f(x)$. Jeżeli dla wprowadzonej wartości x , $f(x)$ nie jest określona, fakt ten jest komunikowany na ekranie. W programie należy zdefiniować i użyć funkcje i procedury:

1. funkcję, która wczytuje wartości x oraz odpowiadające im wartości $f(x)$ z klawiatury. Funkcja zwraca ilość wczytanych wartości par.
2. procedurę, która posortuje rekordy według rosnących wartości x .
3. funkcję zwracającą wartość $f(x)$ dla podanej x .
4. procedurę, która drukuje dane na ekranie.

Przykład:

Podaj x : 3
 Podaj $f(x)$:9
 Podaj x : 2
 Podaj $f(x)$:6
 Podaj x : 1
 Podaj $f(x)$:1
 Podaj x : 8
 Podaj $f(x)$:24
 Podaj x : 0
 Koniec!
 Funkcja:
 $f(3)=9$, $f(2)=6$, $f(1)=1$, $f(8)=24$
 Po sortowaniu: $f(1)=1$, $f(2)=2$, $f(3)=3$, $f(8)=24$
 Podaj x : 1
 $f(1) = 1$;
 Podaj x : 5
 Funkcja nie jest określona dla 5;

21. Z klawiatury wprowadzane są dwa ciągi liczb całkowitych zakończone zerem. Zero nie należy do ciągu; jak zwykle pełni rolę markera końca. Wczytywane ciągi umieszczane są w tablicach jednowymiarowych. Wczytywane wartości nie mogą się powtarzać w obrębie tego samego ciągu! Po wprowadzeniu liczby już występującej w ciągu, program ma zasygnalizować ten fakt komunikatem i powtórzyć wprowadzanie liczby.

Tablice są reprezentacją zbiorów. Różnicą zbiorów Z_1 i Z_2 nazywamy taki zbiór $Z_3 = Z_1 - Z_2$, który zawiera wszystkie te elementy zbioru Z_1 , które nie występują w Z_2 . Oblicz obydwie

różnice wczytanych zbiorów tzn. $Z_1 - Z_2$ i $Z_2 - Z_1$. Wczytane zbiory oraz ich różnice mają być wydrukowane na ekran. W programie należy zdefiniować i użyć następujące funkcje i procedury:

1. funkcja, która wczytuje ciąg z klawiatury i zapisuje ją do tablicy przekazanej jako argument. Funkcja zwraca ilość wczytanych liczb
2. funkcja, która wyznacza różnicę ciągów przekazanych jako argumenty. Funkcja zwraca ilość liczb w wynikowym ciągu.
3. procedura, który drukuje ciąg zawarty w tablicy na ekran.

Przykład:

ciąg A: 3, 5, 7, 1, 8, 34, 21, 16

ciąg B: 97, 8, 3, 34, 16

A-B: 5, 7, 1, 21,

B-A: 97,

22. Z klawiatury wprowadzany jest ciąg liczb całkowitych zakończony zerem. Zero traktowane jest jako znacznik końca ciągu; nie jest elementem ciągu. Wczytany ciąg ma być przechowywany w postaci tablicy rekordów. Rekord składa się z dwóch pól:

- liczba
- ilość jej wystąpień w zbiorze,

Jeśli wczytywana liczba po raz pierwszy pojawia się w ciągu, zajmowany jest dla niej nowy rekord w którym wpisywana jest jej wartość, a ilość wystąpień ustawiana na 1. Jeśli dalej w ciągu nastąpi powtórzenie tej liczby, zamiast zajęcia nowego rekordu, nastąpi inkrementacja licznika jej wystąpień. Po wczytaniu ciągu, program sortuje elementy zbioru według liczby ich wystąpień. Następnie w pętli program prosi użytkownika o podanie z klawiatury liczby całkowitej - wzorca, który zostanie usunięty ze zbioru w następujący sposób:

Jeśli podany liczba znajduje się w zbiorze, program zwraca liczbę jej wystąpień w zbiorze a następnie zmniejsza w rekordzie liczbą wystąpień o jeden. W chwili gdy licznik wystąpień osiągnie wartość zero program ma usunąć rekord z tablicy (pozostałe zaś należy przesunąć tak aby nie było pustych miejsc).

Wczytany zbiór oraz wynik poszukiwań mają być wypisane na ekranie. W programie należy zdefiniować i użyć funkcje i procedury:

1. funkcja, która wczytuje ciąg liczb z klawiatury. Funkcja zwraca ilość wczytanych liczb.
2. procedura, która posortuje zbiór rosnąco.
3. procedura, która dla wczytanego liczby wzorca, usunie jej jedno wystąpienie ze zbioru.
4. funkcje, która dla liczby wzorca zwraca ilość jej wystąpień w zbiorze. Jeśli dana liczba nie występuje w zbiorze, funkcja zwraca -1.
5. procedurę drukującą zbiór na ekran.

Przykład:

Podaj liczbę: 3

Podaj liczbę: 4

Podaj liczbę: 7

Podaj liczbę: 4

Podaj liczbę: 3

Podaj liczbę: 4

Drukuj ciąg:
liczba 3 wystąpiła 2 razy,
liczba 4 wystąpiła 3 razy,
liczba 7 wystąpiła 1 raz,
Po sortowaniu:
liczba 4 wystąpiła 3 razy,
liczba 3 wystąpiła 2 razy,
liczba 7 wystąpiła 1 raz,
Podaj liczbę: 3
Liczebność 3 wynosi: 2
Podaj liczbę: 3
Liczebność 3 wynosi: 1
Podaj liczbę: -3
Liczba -3 nie występuje w zbiorze!

23. Z klawiatury wprowadzany jest ciąg liczb rzeczywistych zakończony zerem. Zero nie należy do ciągu; pełni rolę markera końca. Wczytywany ciąg umieszczany jest w tablicy jednowymiarowej. Program wstępnie sortuje wczytany ciąg w porządku rosnącym. Następnie oblicza różnice kolejno po sobie następujących elementów ciągu (następny minus poprzedni). Obliczone różnice umieszcza w drugiej tablicy. Wczytany ciąg po sortowaniu oraz wynikowy ciąg różnic jest drukowany na ekranie. W programie należy zdefiniować i użyć następujących funkcji i procedur:

1. funkcja wczytująca ciąg z klawiatury. Funkcja zwraca ilość wczytanych liczb.
2. procedura sortująca ciąg w porządku rosnącym dowolną metodą.
3. funkcja obliczająca różnice elementów ciągu. Funkcja zwraca liczbę elementów ciągu wynikowego.
4. procedura, który drukuje ciąg zawarty w tablicy na ekran.

Przykład:

Ciąg: 4.5, 5.0, 7.5, 3.0, -4.0, 6.1
Po sortowaniu: -4.0, 3.0, 4.5, 5.0, 6.1, 7.5
Różnice: 7.0, 1.5, 0.5, 1.1, 1.4

24. Z klawiatury wprowadzany jest ciąg liczb całkowitych zakończony zerem. Zero nie należy do ciągu; pełni rolę markera końca. Wczytywany ciąg umieszczany jest w tablicy jednowymiarowej. Z klawiatury wczytywana jest liczba całkowita - wzorzec. Następnie program rozdziela wczytany ciąg do dwóch tablic, w jednej z tablic umieszcza elementy które są podzielne przez wzorzec, w drugiej pozostałe. Rodzielone tablice są następnie sortowane i wypisywane na ekran. W programie należy zdefiniować i użyć następujących funkcji i procedur:

1. funkcja wczytująca ciąg z klawiatury. Funkcja zwraca ilość wczytanych liczb.
2. procedura sortująca ciąg w tablicy dowolną metodą.
3. procedura rozdzielająca podana tablice do dwóch innych tablic również przekazanych jako argumenty wywołania
4. procedura, który drukuje ciąg zawarty w tablicy na ekran.

Przykład:

Ciąg: 16, 4, 5, 7, 3, 6, 15, 9
Podaj liczbę: 2
Podzielne: 4, 6, 16
Niepodzielne: 3, 5, 7, 9, 15

