

Podstawy Informatyki

Ćwiczenia tablicowe Nr 4

Dr inż. Iwona Oprzędkiewicz

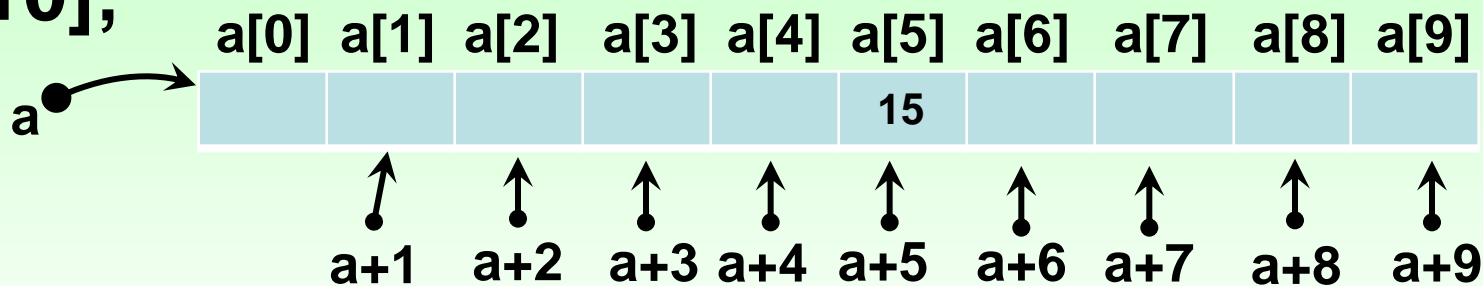
Plan dzisiejszych zajęć:

- 1.Kartkówka z pętli while i do while**
- 2.Programy z zastosowaniem tablic**
- 3.Tematyka następnych zajęć**



I. Typ złożony: tablice

```
int a[10];
```



```
a[5]=15;
```

zapis `a[0]` jest równoważny `*a`
zapis `a[1]` jest równoważny `*(a+1)`
zapis `a[2]` jest równoważny `*(a+2)`

.....

zapis `a[9]` jest równoważny `*(a+9)`

zapis `&a[0]` jest równoważny `a`
zapis `&a[1]` jest równoważny `a+1`
zapis `&a[2]` jest równoważny `a+2`

.....

zapis `&a[9]` jest równoważny `a+9`

czyli jeżeli `x` jest zmienną typu `int`:
`x=a[i]` jest równoważne `x=*(a+i)`

Podsumowując:

`a[k]` i `*(a+k)` to wartości w `k`-tym elemencie tablicy;
`&a[k]` i `a+k` to adresy `k`-tego elementu tablicy



I. Typ złożony: tablice

Przykłady deklaracji tablic:

float liczby[100]; - deklaracja 100-elementowej tablicy liczb zmiennoprzecinkowych

char znaki[25]; - deklaracja 25-elementowej tablicy znaków

int macierz[25][10]; - deklaracja 250- elementowej tablicy dwuwymiarowej liczb int

int *tablica[10]; - deklaracja 10-elementowej tablicy wskaźników do liczb typu całkowitego

char *tekst; - deklaracja wskaźnika do znaku

Przykłady nadawania wartości elementom tablic:

liczby={1,2,10,-5,0,8};

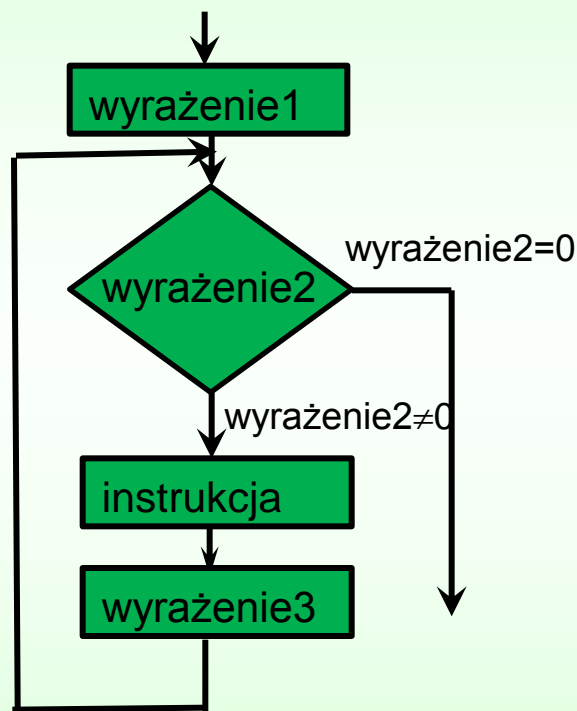
tekst=„wyraz1 wyraz2”;

macierz={{1,2,3},{4,5,6},{7,8,9}};



2. Instrukcja pętli for:

for (wyrażenie1; wyrażenie2; wyrażenie3) instrukcja



wyrażenie1 – określa warunki początkowe pętli

wyrażenie2 – określa warunek wykonania pętli

wyrażenie3 – wykonywane (za każdym razem) po instrukcji w pętli.

Tematy zadań z wykorzystaniem tablic:



1. Napisz program, który przesuwą w lewo o jedną pozycję zawartość tablicy 10-cio elementowej liczb całkowitych tzn. dla elementów $i=0, \dots, 8$ $t[i]$ po wykonaniu programu ma mieć wartość $t[i+1]$, a element $t[9]$ wartość $t[0]$. Program ma używać tylko jednej tablicy.

2. Napisz program, który z dwóch 10-cio elementowych posortowanych tablic liczb zmiennoprzecinkowych tworzy trzecią również posortowaną. Należy wykorzystać fakt posortowania tablic 10-cio elementowych (do trzeciej tablicy, kolejne elementy mają być wstawiane od razu we właściwym miejscu tej tablicy).

3. Napisz program, który z dwóch 10-cio elementowych tablic znakowych tworzy trzecią złożoną na zmianę z elementów to jednej, to drugiej tablicy.

4. Napisz program, który oblicza iloczyn tylko tych elementów całkowitoliczbowej tablicy 10-cio elementowej, które są parzyste.

5. Napisz program, który posortowaną tablicę liczb całkowitych sortuje w kierunku przeciwnym. Należy wykorzystać początkowe posortowanie tablicy. W programie można użyć tylko jedną tablicę.

6. Napisz program, który zamienia liczbę całkowitą dziesiętną na system dwójkowy (bez użycia funkcji matematycznych dostępnych w C i formatu binarnego) .

7. Napisz program, który wpisuje do kolejnych elementów tablicy 20-to elementowej kolejne potęgi liczby n , począwszy od n^0 . Liczbę n należy wczytać z klawiatury. Potęgowanie należy zrealizować za pomocą wielokrotnego mnożenia.

Tematy zadań z wykorzystaniem tablic:

8. Napisz program, który wstawia dowolną liczbę zmiennoprzecinkową do posortowanej tablicy 10-cio elementowej o elementach -5,5; -1,2; 0; 0; 1,4; 3,5; 6,8; 10; 10; 14,5. tak, aby tablica była nadal posortowana bez sortowania całego wektora. Należy zadeklarować tablicę 11-to elementową, żeby wstawiany element się zmieścił.

9. Napisz program, który oblicza średnią arytmetyczną elementów podzielnych przez 15 w 10-cio elementowej tablicy liczb całkowitych.

10. Napisz program, który z dwóch 10-cio elementowych tablic liczbowych tworzy trzecią w ten sposób, że jej kolejne elementy zawierają na zmianę dwa elementy z jednej tablicy i dwa z drugiej. Jeżeli elementy w jednej z tablic skończą się, pozostałe elementy z drugiej są przepisywane.

Struktura programu w C:

```
#include<stdio.h>
```

```
//dyrektywa dla preprocesora, nie jest to element języka C
```

```
#include<math.h>
```

```
//dołączenie innych plików nagłówkowych
```

```
/*<...> tzn. że plik nagłówkowy szukany będzie w katalogu kompilatora  
(najczęściej includes).*/
```

```
#include„abc.h”
```

```
//plik będzie szukany w katalogu, gdzie jest plik z programem źródłowym
```

```
int main()
```

```
{
```

```
//.....
```

```
    return 0;
```

```
}
```


Zasięg zmiennych:

```
int z1,z2;//zmienne globalne deklaruje się przed wszystkimi funkcjami
//specyfikacja C mówi, że ich wartość w chwili deklaracji jest 0
//deklaracja funkcji F0

int main()
{
char a,b,c;//zmienne lokalne dla procedury głównej
/*w języku C powinno się deklarować zmienne na początku bloku (przed pierwszą
instrukcją)*/
/* wg. standardu nowszego C99, dopuszczalna jest deklaracja zmiennej w
dowolnym miejscu bloku (przed jej użyciem)*/
//.....
for(int i=0; i<10;i++) {.....}
// deklaracja zmiennej i obejmuje tylko „blok” instrukcji for
    return 0;
}

//deklaracja funkcji F1
//deklaracja funkcji F2
```

Struktura funkcji w C:

```
typ_funkcji nazwa_funkcji(p1,p2,p3)  
typ_p1 p1; typ_p2 p2; typ_p3 p3;//parametry formalne  
{  
typ1 z1; typ2 z2; typz z;//..... itd..  
//deklaracja zmiennych lokalnych dla funkcji  
//.....ciąg instrukcji wykonywanych w funkcji nazwa_funkcji  
    return z;//może w funkcji występować wiele razy  
}
```

```
//funkcja zwracająca wartość typu float, o dwóch parametrach typu float  
float suma(p1,p2)  
float p1,p2;  
{float x; x=p1+p2; return x;}
```

Uruchomienie funkcji w C:

```
//deklaracja funkcji F0
typ_F0 F0(p1,p2)
typ_p1 p1; typ_p2 p2;
{typ_F0 x;
//.....
return x;
}

int main()
{typ_F1 F1();// zapowiedź
//.....
    return 0;
}

//deklaracja funkcji F1
typ_F1 F1(p1,p2)
typ1_p1 p1; typ1_p2 p2;
{typ_F1 x;
//.....
return x;
}
```

```
//.....

int main()
{
typ_p1 a; typ_p2 b;
typ1_p1 c; typ1_p2 d;
//.....
F0(a,b);// uruchomienie f-cji F0
//.....
F1(c,d);// uruchomienie f-cji F1
//.....
    return 0;
}

//.....
```

Przekazywanie parametrów w C odbywa się przez wartość:

//.....

```
int main()
```

```
{
```

```
typ_p1 a; typ_p2 b;
```

```
typ1_p1 c; typ1_p2 d;
```

```
typ_F0 A; typ_F1 B;
```

```
//.....
```

```
A=F0(a,b);// uruchomienie f-cji F0
```

```
/*do zmiennej A podstawiana jest  
wartość zwracana przez f-cję F0*/
```

```
//.....
```

```
B=F1(c,d);// uruchomienie f-cji F1
```

```
//.....
```

```
return 0;
```

```
}
```

//.....

sterowanie

sterowanie

```
typ_F0 F0(p1,p2)  
typ_p1 p1;  
typ_p2 p2;  
{typ_F0 x;  
//.....  
return x;  
}
```

1. Tworzone są lokalne kopie argumentów a i b jako p1 i p2 (p1 ma wartość a, p2 wartość b).
2. Wykonywane są wszystkie operacje na zmiennych lokalnych (i globalnych, jeżeli takie występują i nie są przesłonięte).
3. return x – przekazuje wartość zmiennej lokalnej x do procedury wywołującej.
4. Pamięć zajęta przez wszystkie zmienne lokalne jest zwalniana

Przykład:

```
int main()
```

```
{float suma();// zapowiedź
```

```
float a,b,c;
```

```
a=11.5; b=2.5;
```

```
c=suma(a,b);
```

```
printf(„suma a+b= %.2f\n”,c);
```

```
printf(„2*%.2f+2*%.2f= %.2f\n”,a,b,suma(a,b));
```

```
return 0;
```

```
}
```

```
//deklaracja funkcji suma
```

```
float suma(p1,p2)
```

```
float p1, p2;
```

```
{float x;
```

```
p1=2*p1;
```

```
p2=2*p2;
```

```
x=p1+p2;
```

```
printf(„p1=%.2f\tp2=%.2f\tsuma=%.2f\n”,p1,p2,x);
```

```
return x;
```

```
}
```

p1=23.00 p2=5.00 suma=28.00

suma a+b= 28.00

p1=23.00 p2=5.00 suma=28.00

2*11.50+2*2.5= 28.00

Co zrobić, jeżeli chcemy jako parametr przekazać do procedury tablicę?

W takiej procedurze parametrem powinien być wskaźnik do zmiennej takiego samego typu, jak elementy tablicy, którą chcemy przekazać do procedury

Np.

```
int main()
{float liczby[10];
int N=5;
//.....
//procedura, która wczytuje tablicę
Czytaj (liczby,N);
//.....
return 0;
}
```

```
void Czytaj(x,m)
float *x;int m;
{
int k;
for(k=0;k<m;k++)
scanf(„%f”,x+k);
}
```

Tematy z funkcji:



1. Napisz program zawierający funkcję, która oblicza symbol Newtona . Program powinien wykonać obliczenia dla 10 zestawów danych (n i k), wczytywanych z podanego przy uruchomieniu pliku (argument programu). Dla każdego zestawu należy wyświetlić liczby i wynik dwumianu.
- 2 . Napisz program, który zawiera funkcję obliczającą średnią arytmetyczną liczb zmiennoprzecinkowych z podanej tablicy (parametr funkcji). W programie należy wykonać funkcję kilkakrotnie dla różnych tablic.
3. Napisz program, który umożliwi wariantowe i wielokrotne wykonywanie następujących kilku funkcji: obliczanie pola prostokąta, obliczanie pola koła, obliczanie pola trójkąta. Program powinien kończyć działanie po naciśnięciu 'K' (należy uwzględnić to w menu w programie).
4. Napisz program zawierający funkcję, która oblicza sumę liczb podzielnych przez 13 w 30 elementowej tablicy liczb całkowitych. Elementy tablicy powinny być wygenerowane losowo z przedziału -50 do 50 w osobnej funkcji. W programie głównym wyświetlić na monitorze wygenerowaną tablicę 30-elementową (odrębna funkcja) oraz wynik funkcji.
5. Napisz program zawierający funkcję, która znajduje (pierwszą jeżeli jest kilka jednakowych) najmniejszą liczbę dodatnią w pierwszych n elementach 30 elementowej tablicy liczb całkowitych. Elementy tablicy powinny być wygenerowane losowo z przedziału -50 do 50 w odrębnej funkcji. W programie głównym wyświetlić na monitorze tablicę n -elementową oraz wynik funkcji.
6. Napisz program zawierający funkcję, która rozpoznaje czy liczba(parametr funkcji) jest pierwsza (zwraca 1 jeżeli jest i 0 jeżeli nie) oraz tworzy tablicę, którą wypełnia 10-cioma kolejnymi liczbami pierwszymi, nie mniejszymi od liczby podanej n (liczba n wczytywana z klawiatury w programie głównym). W programie głównym wyświetlić na monitorze utworzoną tablicę 10-elementową.
7. Napisz program zawierający funkcję zamieniającą całkowitą liczbę dziesiętną na liczbę w systemie szesnastkowym. Należy napisać własną implementację i nie korzystać z gotowych funkcji w C.



Tematy z funkcji:

8. Napisz program zawierający funkcję, która oblicza iloczyn liczb dodatnich, nieparzystych w 50 elementowej tablicy liczb całkowitych. Elementy tablicy powinny być wygenerowane losowo w odrębnej funkcji z przedziału od -100 do 100 oraz wyświetlone na monitorze (w innej funkcji) razem z wynikiem zwróconym przez funkcję.

9. Napisać program sortujący (dowolną metodą) tablicę 30 elementową liczb zmiennoprzecinkowych zawierający funkcję zamiany wartości dwóch zmiennych. Tablica powinna być wypełniana w sposób losowy wartościami z przedziału od -100 do 100 z dokładnością do dwóch miejsc po przecinku (odrębna funkcja).

- Do generowania liczb losowych można użyć funkcję: `rand()`
- Konieczne jest dołączenie pliku nagłówkowego: `stdlib.h`

Następne ćwiczenia:

1. Sprawdzian z tablic.
 2. Tworzenie procedur i funkcji (struktura, zmienne lokalne, przekazywanie parametrów).
- **Uwaga do podanych tematów do przygotowania:**
W napisanych funkcjach nie może być żadnego wczytywania danych z klawiatury . Dane potrzebne funkcji, mogą być przekazywane do funkcji wyłącznie przez parametry funkcji (nie przez zmienne globalne).