

# Laboratorium informatyki

## Programowanie w języku C

### Ćwiczenie 8

#### 1. Wstęp

Zmienne w klasycznej postaci są w stanie przechowywać jedną wartość określonego typu. Tablice umożliwiają przechowywanie pod jedną zmienną określonej liczby indeksowanych wartości danego typu.

#### 2. Tablica

Tablica w odróżnieniu od klasycznej zmiennej umożliwia przechowywanie określonej liczby wartości danego typu. Dane w tablicy indeksowane są od 0 do  $n-1$ , gdzie  $n$  jest wartością całkowitą określającą rozmiar tablicy. Deklarując tablicę należy podać jej typ, nazwę oraz rozmiar. Przy inicjalizacji tablicy nie trzeba deklarować jej rozmiaru. Rozmiar zostanie dobrany automatycznie aby możliwe było przypisanie do tablicy listy zadanych elementów.

```
int tablica_1[5];           // deklaracja 5 elementowej tablicy
int tablica_2[]={1,2,3,4,5}; // inicjalizacja 5 elementowej tablicy
```

Przy deklaracji lub inicjalizacji tablicy przydzielany jest dla niej wymagany rozmiar w pamięci. Aby odwołać się do wybranego elementu tablicy należy podać nazwę tablicy oraz numer indeksu wybranego elementu w nawiasie kwadratowym.

```
tablica_1[0] = 5;
x = tablica_2[2];
```

Deklarując tablicę tworzymy złożoną strukturę danych, pod nazwą której kryje się wskaźnik do obszaru pamięci w którym przechowywana będzie pierwsza jej wartość (o indeksie 0). Indeks umieszczony w nawiasie kwadratowym określa o ile należy zwiększyć wartość wskaźnika aby wskazywał właściwą wartość.

#### 3. Wskaźniki

Wskaźniki są specjalnym typem zmiennych który nie przechowuje wartości (danej) a adres do komórki pamięci w której ta dana jest przechowywana. Wskaźnik więc wskazuje na inną zmienną. Deklarując wskaźnik należy określić jego typ, czyli ustalić na jaki typ danych będzie mógł wskazywać. Wskaźnik deklaruje się analogicznie jak zwykłą zmienną. Nazwę wskaźnika należy poprzedzić znakiem \*. Inicjując wskaźnik można zastosować „deklarator” automatyczny `auto`.

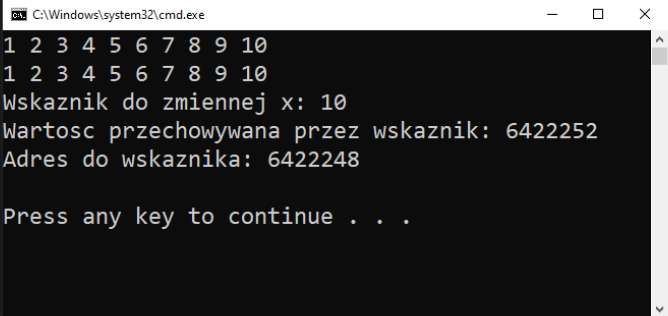
```
int x = 10;
int *wsx;
auto *wx = &x;
```

Aby przypisać do wskaźnika adres zmiennej należy użyć operatora referencji `&`, który „wyciąga” adres zmiennej. Odczytanie wartości na jaką wskazuje wskaźnik realizuje operator wyłuskania `*`.

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     int tablica[] = {1,2,3,4,5,6,7,8,9,10};
7     for(int i=0;i<10;i++)
8         printf("%i ",tablica[i]);
9     printf("\n");
10    for(int i=0;i<10;i++)
11        printf("%i ",*(tablica+i));
12    int x = 10;
13    auto *wX = &x;
14    printf("\n");
15    printf("Wskaznik do zmiennej x: %i\n", *wX);
16    printf("Wartosc przechowywana przez wskaznik: %i\n", wX);
17    printf("Adres do wskaznika: %i\n", &wX);
18 }

```



```

C:\Windows\system32\cmd.exe
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
Wskaznik do zmiennej x: 10
Wartosc przechowywana przez wskaznik: 6422252
Adres do wskaznika: 6422248
Press any key to continue . . .

```

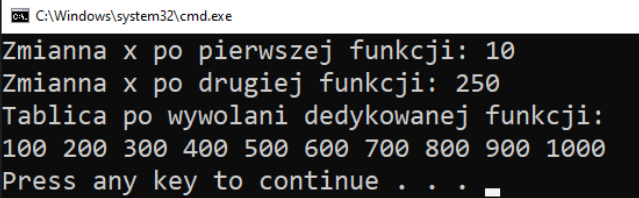
#### 4. Wskaźnik a funkcja

Jednym z podstawowych zastosowań wskaźników jest użycie ich jako parametru wywołania funkcji. Jeżeli parametrem wywołania funkcji jest klasyczna zmienna, to na czas działania funkcji tworzona jest kopia tej zmiennej. Jeżeli zaś parametrem wywołującym jest wskaźnik to w funkcji operujemy na zmiennej bazowej. Oznacza to że z poziomu funkcji możemy zmienić wartość wskazywanej zmiennej. Podobnie jest gdy parametrem wywołania jest tablica. W ten sposób można zbudować funkcję zwracającą więcej jak jedną wartość.

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 void zmianaTab(int tab[]){
4     for(int i=0;i<10;i++)
5         tab[i]=100*tab[i];
6 }
7 void zmianaZmienna2(int *a)
8 {
9     *a=*a*25;
10 }
11 void zmianaZmienna1(int a)
12 {
13     a=a*25;
14 }
15 int main()
16 {
17     int tablica[] = {1,2,3,4,5,6,7,8,9,10};
18     int x = 10;
19     zmianaZmienna1(x);
20     printf("Zmianna x po pierwszej funkcji: %i\n",x);
21     zmianaZmienna2(&x);
22     printf("Zmianna x po drugiej funkcji: %i\n",x);
23     zmianaTab(tablica);
24     printf("Tablica po wywolani dedykowanej funkcji:\n");
25     for(int i=0;i<10;i++)
26         printf("%i ",tablica[i]);
27 }

```



```

C:\Windows\system32\cmd.exe
Zmianna x po pierwszej funkcji: 10
Zmianna x po drugiej funkcji: 250
Tablica po wywolani dedykowanej funkcji:
100 200 300 400 500 600 700 800 900 1000
Press any key to continue . . .

```

#### 5. Zadania

Wykorzystując omówione w instrukcji zagadnienia napisać programy:

- Program sortujący liczby przechowywane w tablicy w kolejności rosnącej.
- Napisać program losujący bez powtórzeń 6 liczb z przedziału 1 – 49. Program wyświetla wylosowane liczby w kolejności rosnącej.