

Podstawy programowania komputerów

Wykład 6: „Tablice, wskaźniki i macierze”



Tablice w programowaniu

Arrays (macierz, tablica) jest to podstawowa struktura danych składająca się z jedno- lub wielowymiarowej tabeli, którą program traktuje jak **jeden obiekt**.

Do każdej danej zapisanej w tablicy można się odwołać przez nazwę tablicy i położenie tej danej wewnątrz tablicy.



Utworzenie tablic

Tablica składa się z ciągu elementów należących do jednego typu danych.

Utworzenie tablicy odbywa się za pośrednictwem deklaracji:

```
float pomiary [100];  
char kod [12];  
int ile [30];
```

Na poszczególne elementy tablicy wskazuje indeks, numeracja elementów rozpoczyna się od zera.



Inicjowanie tablic

Tablicę można inicjować, podając w deklaracji po jej nazwie i znaku równości listę wartości oddzielonych przecinkami i zamkniętych w nawiasach klamrowych.

Jeśli tablica jest inicjowana w deklaracji, to nie jest konieczne podawanie jej rozmiaru.

```
int tab_1[5]={1, 5, 3, 4, 2};
```

```
int tab_2[ ]={1, 5, 6, 3, 4, 5, 6, 77, 19, 0};
```



Użycie tablic w programie

Do tablicy zwracamy się poprzez nazwę zmiennej tablicowej oraz indeks (stała całkowita, zmienna typu całkowitego lub dowolne wyrażenie, którego wynikiem jest liczba całkowita).

Nawiasy zawierające numer elementu tablicy nazywane są operatorem indeksowania.

```
int moja_tab[10];  
int i = 7;  
moja_tab[i] = 10;  
moja_tab[moja_tab[7]-3] = 4;
```



Wskaźnik

W programowaniu jest to zmienna zawierająca odwołanie do innej zmiennej (pojęcie legło u podstaw całej koncepcji języka C).

Utworzenie zmiennej zwanej wskaźnikiem (**pointer variable**) powoduje, iż możemy w pamięci RAM przechowywać adres jakiejś innej zmiennej. Odwołanie się do danych poprzez wskazanie adresu - najszybsza i najefektywniejsza metoda.



Wskaźniki i operatory

Operator adresowy **'&'** przekazuje adres zmiennej:

```
wsk = &zm;
```

Operator dereferencji (pośredniości) **'*'** zwraca wartość przechowywaną przez zmienną:

```
wart = *wsk;
```

```
wart = zm; //ten sam wynik
```

Deklaracja wskaźnika:

```
int * na_wiek;  
char * na_znak;  
float * wsk;
```



Wskaźniki do tablic

Nazwa tablicy **tab_a** jest wskaźnikiem na pierwszy element tablicy:

$$\text{tab_a} \leftrightarrow \&\text{tab_a}[0]$$
$$\text{tab_a}+1 \leftrightarrow \&\text{tab_a}[1]$$

Przykład wykorzystania przy drukowaniu tablicy:

```
for (i=0; i<10; i++)  
    printf("%8d ", *(tab_a+i));
```

i przy zerowaniu tablicy:

```
for (i = 0; i < 10; i++)  
    *(tab_a+i) = 0;
```



Działania na wskaźnikach

- ✓ Zwiększenie wskaźnika można uzyskać przez inkrementację lub dodawanie liczby całkowitej.
- ✓ Odejmowanie wskaźników jest wykonywane w celu określenia, jak daleko od siebie znajdują się dane.

Działania prowadzone są w jednostkach o rozmiarze typu danych: **wsk_1 - wsk_2** \Rightarrow **7** oznacza, że odległość pomiędzy wskazywanymi obiektami w bajtach wynosi 7 razy ilość bajtów przeznaczanych na przechowywanie danego typu danych.



Nazwy tablic jako argumenty

Język C nie pozwala przekazywać całej tablicy jako argumentu funkcji.

Można jednak przekazać funkcji nazwę tablicy (wskaźnik) i rozmiar tablicy (zwykle typ int):

```
int Do_do(int *tab_a, int rozm_a);  
int Do_do(int tab_a [ ], int rozm_a);
```



Ochrona zawartości tablicy

Wykonanie operacji wewnątrz funkcji, która „widzi” tablicę przez argument może zmodyfikować dane w tablicy pierwotnej.

Aby uniemożliwić zmianę zawartości tablicy przez funkcję, w deklaracji argumentu formalnego wystarczy użyć słowa kluczowego **const**:

```
int Do_do(const int *tab_a,  
           int rozm_a);
```



Tablice wielowymiarowe

C nie obsługuje tablice wielowymiarowe wprost, jednak możliwe jest zadeklarowanie **tablicy tablic** (odpowiada tablicy wielowymiarowej):

```
int tablica_2D[3][3];
```

Zainicjowanie tej że tablicy wygląda następująco:

```
int tablica_2D[3][3]={1,1,1,2,2,2,3,3,3};
```

Odwołanie do elementów tablicy dwuwymiarowej następuje poprzez numer tablicy i numer elementu wewnątrz tej tablicy:

```
tablica_2D[1][2] = 10;
```



Zakres tablicy

Kompilator C nie pilnuje zakresu tablic. Dla tego niepoprawną instrukcją jest:

```
tablica_2D[3][1]=6; //maks.indeks tabeli to 2!!!
```

Podana instrukcja kompiluje się poprawnie, wygenerowany kod też nie sprawdza zakresów, więc wykonanie kodu powoduje zapis do pamięci poza zadeklarowaną tablicą.

