

Języki programowania 1

Zapamiętywanie wielu danych tego samego typu – macierze, czyli programowe tablice

Zakres:

- tablice jednowymiarowe,
- indeksowanie elementów tablicy,
- wskaźniki a tablice,
- tablice reprezentujące w C łańcuchy znaków,
- tablice wielowymiarowe,
- tablice bezwymiarowe.

Co to jest tablica?

Czasami przy programowaniu istnieje potrzeba deklaracji wielu zmiennych tego samego typu. Zamiast je wszystkie osobno deklarować, można zadeklarować cały zestaw zmiennych o tym samym typie. Deklaracja taka nazywa się tablicą lub macierzą.

Każda tablica jest zbiorem elementów jednego typu. Każda zmienna wchodząca w skład tablicy danego typu nazywana jest elementem tablicy. Do wszystkich elementów możemy odwoływać się posługując się nazwą tablicy. Wszystkie elementy tablicy są przechowywane w pamięci razem – w kolejnych komórkach pamięci.

Ogólna postać deklaracji tablicy jest następująca:

```
typ_danych Nazwa_tablicy[rozmiar_tablicy];
```

gdzie:

- typ_danych, typ wszystkich zmiennych wchodzących w skład tablicy,
- Nazwa_tablicy, identyfikator nadawany zgodnie ze standardami,
- rozmiar_tablicy, określa ile elementów zawiera tablica.

Para nawiasów prostokątnych [...] bezpośrednio informuje kompilator, że jest to deklaracja tablicy. Przykładowo tablica jednowymiarowa ośmioelementowa może być zadeklarowana w następujący sposób:

```
int tablica[8];
```

Liczba 8 określa rozmiar tablicy i oznacza, że w danej tablicy może być najwyżej 8 elementów typu całkowitego. Podobnie jak wszystkie zmienne tablica musi być najpierw zadeklarowana.

Indeksowanie elementów tablic

Po zadeklarowaniu tablicy, można przez nią odwoływać się do każdego jej elementu. Przykładowo:

```
char dzien[7];
```

oznacza zadeklarowanie siedmioelementowej tablicy typu znakowego. Elementy w tablicy są ustawione szeregowo (jeden za drugim). Należy pamiętać, że pierwszy element tablicy to zawsze jest 0 (dzien[0]). Do kolejnych elementów możemy się odwoływać jako: dzien[1], dzien[2],...dzien[6]. Odwołania te nazywane są referencjami do elementów tablicy (ponieważ umożliwiają odwołanie się do danych jej elementów).

Inicjowanie tablic

Posługując się danymi wyrażeniami referencyjnymi można zainicjować każdy element tablicy z osobna. Przykładowo:

```
dzien[0]='P';
```

Podobnie można zadeklarować wszystkie elementy tablicy:

```
int dane[5]={50,4,67,12,5};
```

Dane zadeklarowane w ten sposób będą w następujący sposób umieszczone w pięcioelementowej tablicy dane typu całkowitego:

```
int dane[0]=>50
```

```
int dane[1]=>4
```

```
int dane[2]=>67
```

```
int dane[3]=>12
```

```
int dane[4]=>5
```

Przykład inicjowania tablicy jednowymiarowej – *tablica.c*.

Rozmiar tablicy

Poszczególne elementy tablicy rozmieszczane są w kolejnych komórkach pamięci. Cała tablica zajmuje ciągły obszar pamięci. Chcąc obliczyć całkowitą ilość bajtów pamięci zajmowanych przez daną tablicę można obliczyć:

```
sizeof(typ_danych)*rozmiar_tablicy
```

Zwrócony wynik będzie całkowitą ilością bajtów w pamięci, którą zajmuje dana tablica. Drugim sposobem wyznaczenia całkowitego rozmiaru zajmowanego przez tablicę jest bezpośrednia deklaracja:

```
sizeof(nazwa_tablicy);
```

Przykład obliczania wielkości tablicy w bajtach – *rozmiar.c*.

Wskaźniki a tablice w C

Tablice i wskaźniki w C mają ścisły związek. Aby wskaźnik wskazywał adres pierwszego elementu tablicy, to wystarczy go zainicjować posługując się nazwą tablicy poprzez zwykłe przypisanie. Jeżeli dany wskaźnik wskazuje tablicę, to poprzez niego można w bardzo prosty sposób odwoływać się do poszczególnych elementów tablicy. Przykład zawarty jest w programie *odwołanie.c*.

Drukowanie zawartości tablic znakowych

Dość często się zdarza, że programując tworzymy tablice znakowe. W większości komputerów dane char zajmują 1 bajt w pamięci. Zatem jedna komórka tablicy zajmuje również jeden bajt. Całkowita ilość elementów zatem, będzie równa wielkości tablicy w bajtach. Dlatego w C łańcuchy znaków (teksty) określane są jako jednowymiarowe tablice tekstowe.

Przykład drukowania zawartości tablicy znakowej przedstawiony jest w programie *druk.c*.

Znak zerowy - \0

Znak zerowy \0 jest w C traktowany jako pojedynczy znak. Jest to znak specjalny, zaznaczający koniec łańcucha znakowego. Dlatego właśnie, gdy funkcja printf() przetwarza łańcuch znaków, wyprowadza znaki jeden po drugim, aż do momentu napotkania znaku specjalnego. Znak specjalny \0 jest oceniany zawsze jako logiczny fałsz.

Tablice wielowymiarowe

Do tej pory tablice, które były omawiane były tablicami jednowymiarowymi. W języku C istnieje oczywiście możliwość wprowadzania tablic

wielowymiarowych (tzw. macierze wielowymiarowe). Uogólniony format deklaracji tablicy N-wymiarowej jest następujący:

Typ_danych Nazwa_tablicy[rozmiar_tablicy1] [rozmiar_tablicy2] ... [rozmiar_tablicyN]

N- jest dodatnie i całkowite.

Przykładowa deklaracja tablicy dwuwymiarowej:

int tablica[2][3];

Zainicjować taką tabelę można element po elemencie:

tablica[0][0]=1;

tablica[0][1]=2;

tablica[0][2]=3;

tablica[1][0]=4;

tablica[1][1]=5;

tablica[1][2]=6;

lub też w następujący sposób:

int tablica[2][3]={1,2,3,4,5,6};

int tablica[2][3]={{1,2,3},{4,5,6}};

Przykład wydruku tablicy dwuwymiarowej program – *tab_dwa.c*.

Tablice bezwymiarowe

Nie zawsze deklarując tablicę musimy wpisywać w nawiasy kwadratowe rozmiar. Kompilator może automatycznie obliczyć rozmiar tablicy. Jeżeli przykładowo kompilator napotka w kodzie zapis:

int tablica[]={1,2,3,4,5,6,7,8};

utworzy tablicę tak dużą, aby zmieścić wszystkie jej elementy. W podobny sposób można zadeklarować tablice wielowymiarowe, ale pozostawiając nieokreślony pierwszy rozmiar wielowymiarowej tablicy. Wówczas tylko zapis ten będzie jednoznaczny. Przykładowo:

```
char lista[][2];
```

Niejednoznaczny będzie zapis:

```
char lista[][] (?!?)
```