
Politechnika Świętokrzyska
Wydział Elektrotechniki, Automatyki i Informatyki
Katedra Elektrotechniki Przemysłowej i Automatyki
Zakład Urządzeń i Systemów Automatyki

Programowanie komputerów 2

Tablice

Instrukcja laboratoryjna
(wersja robocza)

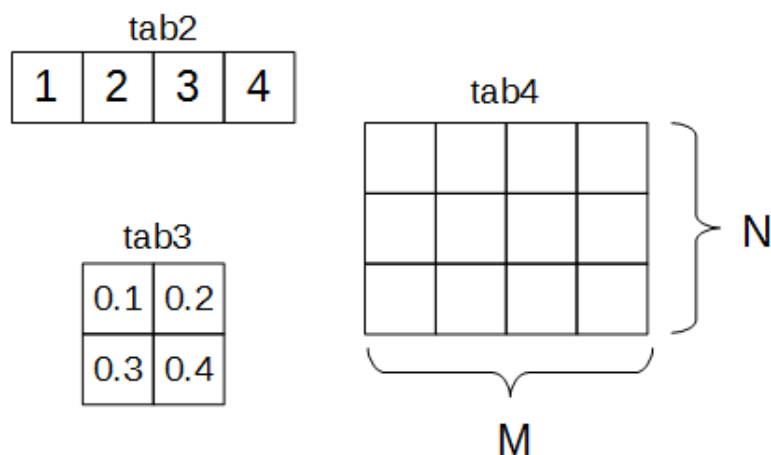
Paweł Strączyński
2017

Tablica to zbiór zmiennych jednego typu. W języku C wyróżniamy tablice jedno- i wielowymiarowe. Tablice jednowymiarowe nazywa się wektorami, dwuwymiarowe natomiast macierzami. Celem stosowania tablicy jest zastąpienie wielu zmiennych tego samego typu jednym zbiorem elementów. Na listingu poniżej przedstawiono sposób deklarowania tablic.

Listing 1. Deklaracja tablic w języku C

```
#define N 3
#define M 4
int tab1[5]; //deklaracja jednoelementowej tablicy o pięciu elementach
int tab2[] = {1, 2, 3, 4}; //deklaracja zainicjowanej tablicy jednowymiarowej
float tab3[][] = {{0.1, 0.2}, {0.3, 0.4}}; //deklaracja zainicjowanej
//tablicy dwuwymiarowej
char tab4[N][M]; //deklaracja dwuwymiarowej tablicy typu znakowego
```

Jak widać tablica podczas deklaracji może zostać zainicjowana wartościami.



Rysunek 1. Tablice przykładowe

Na listingu poniżej przedstawiono sposób dostępu do elementów tablicy. Należy zaznaczyć iż indeksowanie elementów tablic w języku C zaczyna się od 0.

Listing 2. Przykład dostępu do elementów tablicy

```
int i;
int tab[5];
for(i=0; i < sizeof(tab) / sizeof(tab[0]); i++)
{
    tab[i]=i*2+1;
    printf("Element %d tablicy zawiera wartosc %d\n",i,tab[i]);
}
```

Wynik działania programu z listingu 2 przedstawiono poniżej.

Listing 3. Wynik działania programu z listingu 2

```
Element 0 tablicy zawiera wartosc 1
Element 1 tablicy zawiera wartosc 3
Element 2 tablicy zawiera wartosc 5
Element 3 tablicy zawiera wartosc 7
Element 4 tablicy zawiera wartosc 9
```

Na listingu 2 przedstawiono sposób wyznaczenia elementów tablicy. Należy podzielić rozmiar całej tablicy przez rozmiar jednego z jej elementów (najlepiej pierwszego o indeksie 0) z wykorzystaniem operatora `sizeof`.

W języku C nie ma specjalnego typu do przechowywania tekstu (string). Łańcuchy znakowe można przechowywać w tablicach z elementami typu `char`. Tablica zawierająca łańcuch znakowy zawiera o jeden element więcej niż liczba znaków tekstu który przechowuje. Ostatnim elementem takiej tablicy jest znak `'\0'` który oznacza koniec ciągu znakowego. Przykład tablic przechowujących ciągi znaków przedstawiono poniżej.

Listing 4. Deklarowanie łańcuchów znakowych

```
char text1[] = {'F','o','o','b','a','r','\0'};
char text1[] = "Foobar";
```

Inicjując tablicę przypisując do niej tekst podany w cudzysłowie nie należy dodawać znaku `'\0'` gdyż zostanie dodany automatycznie.

Załączając do programu plik nagłówkowy `string.h` dołącza się szereg funkcji pozwalających dokonywać różnego typu operacje na łańcuchach znakowych. Kilka przydatnych funkcji przedstawiono w tabeli 1.

Tabela 1. Funkcje pozwalające na dokonywanie operacji na łańcuchach znakowych

Deklaracja funkcji	Opis
<code>int strlen(char *str)</code>	Zwraca długość ciągu znakowego
<code>int strncmp(char *str1, char *str2, n)</code>	Porównuje n znaków dwóch ciągów znakowych. Jeżeli łańcuchy są takie same zwraca 0
<code>strcpy(char *str1, char *str2);</code>	Kopiuje łańcuch będący drugim argumentem do tablicy podanej jako pierwszy argument
<code>strcat(str1, str2, n);</code>	Dołącza do ciągu znaków podanego jako pierwszy argument zawartość drugiego argumentu (ciąg znaków)

Zadania do wykonania

Zadanie 1

Napisać program który wyświetli na standardowym wyjściu zawartość N-elementowej tablicy liczb pseudolosowych. Ilość elementów N powinna być zdefiniowana w programie przy użyciu stałej. Do generowania liczb pseudolosowych użyć funkcji `rand()`.

Wskazówka:

Przed użyciem funkcji `rand()` pamiętać o zainicjowaniu generatora liczb pseudolosowych – `srand(time(NULL))`. Dołączyć plik nagłówkowy `time.h`

Zadanie 2

Napisać program obliczający wyznacznik macierzy o rozmiarze 2x2. Zawartość poszczególnych elementów macierzy przechowywać w tablicy dwuwymiarowej. Wartość poszczególnych elementów macierzy powinna być wprowadzana za pomocą klawiatury.

Wskazówka:

$$\det(\mathbf{A}) = \det \left(\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \right) = a_{11} \cdot a_{22} - a_{12} \cdot a_{21}$$

Zadanie 3

Napisać program dokonujący dodawania dwóch macierzy o rozmiarze podanym przez użytkownika. Maksymalne wymiary macierzy powinny być zdefiniowane w postaci stałych. Program powinien:

- pobierać ze standardowego wejścia wymiary macierzy (jeżeli przekraczają maksymalne wymiary określone w programie stałymi powinien zostać wyświetlony stosowny komunikat),
- pobierać ze standardowego wejścia wartości poszczególnych elementów macierzy,
- drukować na standardowym wyjściu zawartość wprowadzonych macierzy oraz wynik operacji.

Zadanie 4

Rozbudować program z zadania 3 o funkcjonalność mnożenia macierzy.