
Podstawy programowania.

Wykład 10

Tablice wielowymiarowe, tablice wskaźników

Tablice wielowymiarowe

- C umożliwia definiowanie tablic wielowymiarowych
 - najczęściej stosowane są tablice dwuwymiarowe
 - matematycznym odpowiednikiem tablic dwuwymiarowych są macierze
- Definicja tablicy k-wymiarowej ma postać:
 - `typ nazwa[N_1][N_2]..[N_k];` // + ewentualne inicjowanie, np.
`int tab_int_2D[2][3] = { {0,1,2}, {4,5,6} };`
 - np. dla tablicy znaków przechowującej wiele linii tekstu:
`#define MAX_DL_L 55` // maksymalna liczba znaków w jednej linii tekstu
`#define MAX_N_L 15` // maksymalna liczba linii tekstu
`char tekst_wieloliniowy[MAX_N_L][MAX_DL_L];`
- Tablice wielowymiarowe przechowywane są w kolejnych komórkach pamięci
 - istnieje tylko jeden symbol (nie zmienna) w programie zawierający adres tablicy wielowymiarowej

Tablice wielowymiarowe

→ Dostęp do elementu tablicy wykorzystuje notację:

```
int linia = 5; int znak_w_linii = 27;  
tekst_wieloliniowy[ linia ][ znak_w_linii ] = 'q';  
printf("%c", tekst_wieloliniowy[ 5 ][ 27 ]);
```

→ Standardowy sposób przechowywania tablic dwuwymiarowych w C oznacza przechowywanie macierzy wierszami

- indeksy elementu tablicy służą do prostego obliczenia jego adresu:

$\text{adres} = \text{początek} + (\text{linia} * \text{MAX_DL_L} + \text{znak_w_linii}) * \text{rozmiar_typu}$

- z tego względu tablice wielowymiarowe często zamieniane są na tablice jednowymiarowe:

```
char tekst_wieloliniowy_1D[ MAX_N_L * MAX_DL_L ];  
int linia = 5; int znak_w_linii = 27;  
tekst_wieloliniowy_1D[ linia * MAX_DL_L + znak_w_linii ] = 'q';  
printf("%c", tekst_wieloliniowy_1D[ 5 * MAX_DL_L + 27 ]);
```

Tablice wielowymiarowe

- Przesyłanie tablic wielowymiarowych jako argumentów funkcji realizuje prosta notacja:

```
funkcja_tab_wielo( nazwa_tab_wielo );
```

```
wczytaj_tekst ( tekst_wieloliniowy );
```

- Każdorazowo wewnątrz funkcji musi być możliwe obliczenie adresu dowolnego elementu posługując się indeksami elementu, jak np. w przypadku przykładowej tablicy dwuwymiarowej:

```
adres = początek + ( linia*MAX_DL_L + znak_w_linii ) * rozmiar_typu
```

- oznacza to, że poza nazwą tablicy (czyli adresem jej początku) muszą zostać przekazane dodatkowe parametry określające rozmiar tablicy, np. w przypadku przykładowej tablicy dwuwymiarowej parametr MAX_DL_L
- parametr ten musi znaleźć się w deklaracji funkcji, co oznacza, że musi być znany w trakcie kompilacji! (w klasycznym C)

Tablice wielowymiarowe

- Deklaracje funkcji przyjmujących jako argument tablicę wielowymiarową mogą mieć jedną z równoważnych postaci:

```
wczytaj_tekst ( char tekst_wieloliniowy [N_L] [MAX_DL_L] );
```

```
wczytaj_tekst ( char tekst_wieloliniowy [ ] [MAX_DL_L] );
```

```
wczytaj_tekst ( char (*tekst_wieloliniowy)[MAX_DL_L] );
```

- ostatnia postać wykorzystuje utożsamienie nazwy tablicy ze wskaźnikiem
- W przypadku alternatywnego przechowywania macierzy w postaci tablic jednowymiarowych:
- parametry (wymiary) tablicy nie muszą być znane w czasie kompilacji – muszą jednak zostać w jakiś sposób przekazane funkcji
 - deklaracja funkcji:

```
wczytaj_tekst_1D( char* tekst_wieloliniowy, int max_dl_l);
```
 - wywołanie:

```
wczytaj_tekst_1D( tekst_wieloliniowy, max_dl_l);
```

Tablice wskaźników

- Tablice wielowymiarowe nie są elastycznym i często używanym elementem C
- Znacznie częściej stosowane są bardziej elastyczne tablice wskaźników, np. dla poprzedniego przykładu:

```
#define MAX_N_L 15 // maksymalna liczba linii tekstu
```

```
char* tekst_wieloliniowy_tab_wsk[MAX_N_L] = {NULL}; // notacja!
```

- wskaźniki są zainicjowane dla bezpiecznego użycia
- ich wartości muszą zostać podstawione jako adresy istniejących zmiennych lub za pomocą funkcji alokacji pamięci, np.

```
char napis_tab[10] = "..."; tekst_wieloliniowy_tab_wsk[0] = napis_tab;
```

```
int dl_l = 25; int linia = 5;
```

```
tekst_wieloliniowy_tab_wsk[linia] = malloc( dl_l*sizeof(char));
```

- w powyższym przypadku `tekst_wieloliniowy_tab_wsk` jest symbolem oznaczającym tablicę (jej adres), natomiast składowe tej tablicy są zmiennymi, które mogą przyjmować różne wartości (adresów)

Tablice wskaźników

- Wskaźniki będące składowymi tablicy wskaźników mogą pokazywać na rozmaite obiekty w pamięci
 - wskaźniki muszą zachować zgodność typu wskazywanych obiektów z deklaracją tablicy:

```
int* tab_wsk_int[10] = {NULL};
```

```
double* tab_wsk_d[10] = {NULL};
```

```
char* tekst_wieloliniowy_tab_wsk[MAX_N_L] = {NULL};
```

- Dostęp do obiektów wskazywanych przez elementy tablicy wskaźników zależy od tego na co wskazują elementy:

- elementy mogą pokazywać na pojedyncze zmienne, np.:

```
int liczba_int=5; int* tab_wsk_int[10] = {NULL};
```

```
tab_wsk_int[0] = &liczba_int;
```

```
printf("%d", *tab_wsk_int[0]); // lub printf("%d", tab_wsk_int[0][0]);
```

- taki sposób daje możliwość dostępu do pojedynczej zmiennej z wielu miejsc (funkcji, struktur) w programie

Tablice wskaźników

- Dostęp do obiektów wskazywanych przez elementy tablicy wskaźników zależy od tego na co wskazują elementy:
- elementy mogą pokazywać na tablice zmiennych:

```
double tab_d[10] = {1.0, 2.0, 0};  
double* tab_wsk_d[10] = {NULL};  
tab_wsk_d[0] = tab_d;  
printf("%lf", tab_wsk_d[0][1]);
```

 - taki sposób użycia daje elastyczność stosowania – wskazywane tablice mogą mieć różną długość (jak w poprzednim przykładzie)
 - elementy mogą pokazywać na struktury

```
typedef struct mg_punkt_2D{ double x; double y; } mg_punkt_2D;  
mg_punkt_2D* koniec[2] = {NULL}; mg_punkt_2D p1 = {0.0,0.0};  
koniec[0] = &p1; koniec[0]->x = 1.0;  
printf("%lf\n", koniec[0]->x);
```

 - tablice wskaźników do struktur mogą służyć do konstruowania złożonych struktur danych

Tablice wskaźników

→ Przesyłanie tablic wskaźników jako argumentów funkcji wykorzystuje notacje:

- deklaracji funkcji (2 możliwe warianty):

```
wypisz_tekst( char* tekst_wieloliniowy_tab_wsk[] );
```

```
wypisz_tekst( char** tekst_wieloliniowy_tab_wsk );
```

- notacja `char**` zwraca uwagę na fakt, że na stosie funkcji `wypisz_tekst` będzie znajdowała się zmienna będąca wskaźnikiem do wskaźnika – czyli wskaźnikiem do pierwszego elementu tablicy wskaźników)

- wywołania funkcji

```
wypisz_tekst( tekst_wieloliniowy_tab_wsk );
```

- wewnątrz funkcji musi istnieć sposób poprawnego realizowania dostępu do obiektów wskazywanych przez elementy tablicy
 - dla tablic napisów można wykorzystać fakt istnienia `'\0'` na końcu napisu