Laboratorium informatyki

Programowanie w języku C

Ćwiczenie 9

1. Wstęp

Tablice w języku C mogą być także wielowymiarowe. Oznacza to że można zadeklarować i korzystać z tablic dwu, trzy i więcej wymiarowych. Tablice można deklarować, podobnie jak zmienne w języku C w dowolnym miejscu kodu. Z tego względu możliw2e jest dynamiczne zadeklarowanie tablicy, której rozmiar nie jest podany jawnie, a zależy od wartości zmiennej.

2. Tablica

Inicjują tablicę w języku C musimy podać informację o jej rozmiarze lub podając listę elementów tablicy. Możliwe jest także zainicjowanie tablicy bez podawania jej rozmiaru i przypisywania pełnej listy elementów. Można przypisać tylko wybrane elementy tablicy, a jej rozmiar będzie równy największemu indeksowi przypisanego elementu.

```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int tab[] = {[0]=1,[5]=9};
    for(int i=0;i<6;i++)
        printf("%i:\t%i\n",i,tab[i]);

}
```

```
0:      1
1:      0
2:      0
3:      0
4:      0
5:      9

Press any key
```

3. Tablice parametryczna

W języku C można zdeklarować tablicę z sparametryzowana informacją o liczbie elementów. Zmienna określająca rozmiar musi być jednak określona w momencie zdeklarowania tablicy.

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    srand(time(0));
    int s = rand()%10+1;
    int tab[s];
    printf("Rozmiar tablicy=%i\n",s);
    for(int i=0;i<s;i++){
        tab[i] = rand()%101;
        printf("%i:\t%i\n",i,tab[i]);
    }
}
```

```
Rozmiar tablicy=9
0:      76
1:      65
2:      83
3:      65
4:      61
5:      42
6:      4
7:      14
8:      61

Press any key to c
```

4. Tablice 2D

Tablica wielowymiarowe wymagają w momencie deklaracji określenia rozmiaru każdego wymiaru lub przy inicjalizacji podania pełnej listy przypisanych elementów.

```
int tab[liczba wierszy][liczba kolumn];
```

```
int tab[5][5] = {{0}};
```

Przy deklaracji tablicy konieczne jest podanie liczby kolumn. Inicjując niepełną tablicę 2D pozostałe elementy zapełniane są zerami.

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
void printTab(int tab[3][3])
{
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            printf("%i ",tab[i][j]);
        }
        printf("\n");
    }
    printf("---------------\n");
}
int main()
{
    int tab1[3][3];
    int tab2[3][3] = {1,2,3,1,2,3,1,2,3};
    int tab3[3][3] = {1,2,3,1,2};
    int tab4[][3] = {{1,2,3},{1,2,3},{1,2,3}};
    int tab5[3][3] = {[2][2]=10};
    int tab6[][3] = {1,2,3,1,2,3,1,2,3};

    printTab(tab1);
    printTab(tab2);
    printTab(tab3);
    printTab(tab4);
    printTab(tab5);
    printTab(tab6);
}
```

```
C:\Windows\system32\cmd.exe
6422280 1999925037 4201408
6422400 4201520 4201408
4199120 2760704 4194432
---------------
1 2 3
1 2 3
1 2 3
---------------
1 2 3
1 2 0
0 0 0
---------------
1 2 3
1 2 3
1 2 3
---------------
0 0 0
0 0 0
0 0 10
---------------
1 2 3
1 2 3
1 2 3
---------------
```

5. Tablice 2d a funkcje

Podobnie jak w przypadku tablic jednowymiarowych tablice 2D także mogą być parametrami funkcji. Podobnie jak i przy tablicach jednowymiarowych przy zapisie kodu funkcji uniwersalnej konieczna jest parametryzacja rozmiaru tablicy i przekazanie go jako jeden z parametrów funkcji. Przy zapisywaniu deklaracji funkcji z tablicą 2D jako parametrem konieczne jest podanie przynajmniej liczby kolumn tablicy jako wartości lub poprzez parametr. Parametr określający rozmiar tablicy musi w deklaracji być umieszczony przed deklaracją tablicy. Należy pamiętać że tablica przekazywana jest do funkcji jako wskaźnik i nie można określić jej rozmiaru (liczby elementów) wewnątrz funkcji za pomocą funkcji `sizeof()`.

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
void printTab(int n,int tab[n][n])
{
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            printf("%i ",tab[i][j]);
        }
        printf("\n");
    }
}
void genrator(int n, int m, int tab[n][m])
{
    for(int i=0;i<n;i++)
        for(int j=0;j<m;j++)
            tab[i][j] = rand()%10;
}
int main()
{
    srand(time(0));
    int tab[10][10]={{0}};
    int s = sizeof(tab)/sizeof(int);
    s = sqrt(s);
    printf("Rozmiar tablicy: %i\n",s);
    printTab(s,tab);
    printf("---------------\n");
    genrator(s,s,tab);
    printTab(s,tab);
}
```

```
C:\Windows\system32\cmd.exe
Rozmiar tablicy: 10
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
---------------
1 4 6 6 3 6 4 0 0 1
0 9 6 5 7 2 8 8 9 0
5 8 7 8 5 5 7 8 8 0
5 6 6 2 7 4 0 9 3 3
2 2 5 6 3 2 5 7 2 0
3 7 9 2 0 4 0 8 4 5
9 9 9 3 5 5 8 0 0 3
9 2 9 2 7 3 2 2 2 3
0 7 8 3 7 6 4 0 2 2
2 1 9 1 7 5 3 6 4 1

Press any key to cont
```

6. Tablica dynamiczna

Wielkość tablicy deklarowanej w języku C jest z góry ustalona i nie może być zmieniona. Jednakże ze względu na sposób przechowywania danych tablicy w pamięci i metody odwoływania się do tych wartości możliwe jest dynamiczne zarządzanie rozmiarem tego obszaru, a więc i wielkością tablicy. Nazwa każdej zadeklarowanej tablicy jest wskaźnikiem na początek obszaru pamięci przydzielonego dla tablicy.

Dynamiczne alokowanie polega na zarezerwowaniu obszaru pamięci który może pomieści określoną ilość danych i przypisanie adresu na początek tego obszaru do wskaźnika. Zadanie to realizują funkcje `malloc()` i `calloc()` z biblioteki `<stdlib.h>`. Obydwie wykonują powyżej opisaną czynność, a funkcja `calloc()` dodatkowo ustawia wartość każdego elementu tablicy na 0. Zmiana rozmiaru tablicy dynamicznej realizowana przez funkcję realloc() możliwa jest zarówno w celu jej powiększenia jak i pomniejszenia. Funkcja ta tworzy nową tablicę dynamiczną o zadeklarowanym w niej rozmiarze i przenosi dane z tablicy źródłowej, zwracając adres początku obszaru nowej tablicy. Przy dynamicznej alokacji należy pamiętać o procedurze zwolnienia pamięci zajmowanej przez dynamiczną tablicę. W odróżnieniu od zwykłej tablicy po zakończeniu programu kasowany jest wyłącznie wskaźnik i zwalniania pamięć przez niego zajmowana, czyli

adres na początek obszaru tablicy. Sam ten obszar nie jest zwalniany. Należy to zrobić instrukcją `free()`. Przykład dynamicznej alokacji tablicy pokazano na poniższym przykładzie.

```c
#include <stdlib.h>
#include <stdio.h>
void drukuj(int *T, int s)
{
    printf("Tablica<0x%x>:",T);
    for(int i=0;i<s;i++)
        printf("%i ",T[i]);
    printf("\n--------------------\n");
}
int main()
{
    int *tab1,*tab2;
    tab1 = malloc(3*sizeof(int));
    tab2 = calloc(3,sizeof(int));
    drukuj(tab1,3);
    drukuj(tab2,3);
    tab2 = realloc(tab2,5*sizeof(int));
    drukuj(tab2,5);
    free(tab1);
    free(tab2);
}
```

```
C:\WINDOWS\system32
Tablica<0x71c713f0>:1908892704 534 1908867408

--------------------
Tablica<0x71c71410>:0 0 0
--------------------
Tablica<0x71c71410>:0 0 0 0 0
--------------------

Press any key to continue . . .
```

Dynamiczna alokacja pamięci dla tablicy wielowymiarowej także jest możliwa i realizowana w analogiczny sposób który wynika z sposobu osadzenia tablicy wielowymiarowej w pamięci. Proces ma o jeden etap więcej jak liczba wymiarów tablicy:

a) Utworzenie wskaźnika stopnia zgodnego z liczbą wymiarów tablicy (dla tablicy 2D → `typ **tab2D;`, dla tablicy 3D→ `typ ***tab3D;`
b) Utworzenie tablicy wskaźników dla pierwszego wymiaru (`tab2D = malloc(n*sizeof(typ*)); tab3D = malloc(n*sizeof(typ**));`)
c) Dla każdego wskaźnika w tablicy utworzonej w punkcie b) tworzy się tablicę kolejnego wymiaru. Zadanie to realizuje się w pętli iteracyjnej dla wszystkich elementów tablicy wyższego wymiaru. Czynność ta jest realizowana w kaskadzie w zależności od liczby wymiarów

Zwalnianie pamięci należy realizować w kolejności odwrotnej.

7. Zadania

Wykorzystując omówione w instrukcji zagadnienia napisać:

A. Funkcję generującą tablicę jednowymiarową o zadanej liczbie elementów i zawierającą liczby losowe z podanego zakresu wartości.
B. Napisać funkcje zwracającą tablicę 2D kwadratową zawierającą wartości losowe z zadanego zakresu liczb.

C. Napisać funkcję zwracającą tablicę 2D zawierającą wartości losowe z zadanego zakresu liczb, w której liczba elementów każdego wiersza jest ustalana losowo w zakresie od 1 do podwojonej wartości liczby wierszy.
D. Napisać funkcję zmieniającą rozmiar tablicy 2D z przykładu B i C .
E. Napisać funkcję wyświetlającą zawartość tablicy 2D.
F. Napisać program demonstrujący utworzone funkcje.