Podstawy Programowania
Zajęcia laboratoryjne 5
I rok Bioinformatyki Politechniki Poznańskiej

1 Tablice

Tablica to ciąg zmiennych jednego typu, gdzie do poszczególnych elementów można się odwołać poprzez numer indeksu (numery indeksów rozpoczynają się od 0). Przy deklaracji tablicy należy określić jej typ, nazwę oraz rozmiar, różne sposoby deklaracji tablicy:

- typ nazwa[rozmiar] - najczęściej używana deklaracja tablicy,
- typ nazwa[rozmiar] = {wartosc_1, wartosc_2, ...} - deklaracja tablicy wraz z określeniem jej zawartości,
- typ nazwa[] = {wartosc_1, wartosc_2, ...} - deklaracja tablicy wraz z określeniem jej zawartości nie wymaga podania rozmiaru.

Przykład deklaracji tablicy przedstawiono poniżej (kod 1 dostępny w materiałach na stronie):

```c
#include <stdio.h>
int main()
{
    int tab[10]; //okreslenie typu, nazwy zmiennej tablicowej i rozmiaru tablicy
    tab[0] = 1; //przypisanie wartosci 1 do zerowego elementu tablicy

    for(int i = 0; i < 10; i++)
    {
        printf("Wartosc: %d \n", tab[i]); //wyswietlenie zawartosci tablicy
    }
    return 0;
}
```

Zastanów się co dzieje się powyższym kodzie i dlaczego? Zwróć uwagę, że w kodzie nie przypisano wartości do poszczególnych elementów tablicy, z wyjątkiem elementu 0 (tab[0] = 1).

Przykład przypisania wartości do tablicy w pętli (kod 2 dostępny w materiałach na stronie):

```c
#include <stdio.h>
int main()
{
    int n;
    printf("podaj ilosc liczb: ");
    scanf ("%d", &n);

    int tab[n];

    for (int i = 0; i < n; i++)
    {
        tab[i] = i; //przypisanie i'temu elementowi tablicy wartosci i
        printf("%d\n", tab[i]);
```

```
14      }
15      return 0;
16  }
```

Operator sizeof służy do poznania wielkości tablicy, dokładnie operator ten zwraca cały rozmiar pamięciowy tablicy. Natomiast sizeof *tab zwraca rozmiar typu elementu tablicy. W związku z tym dzieląc cały rozmiar pamięciowy tablicy przez rozmiar pojedynczego elementu uzyskujemy liczbę elementów tablicy. Poniżej przykładowy kod z użyciem sizeof (kod 3 dostępny w materiałach na stronie):

```
1  #include <stdio.h>
2  int main()
3  {
4      int tab[5] = {1,2,3,4,5};
5
6      printf("sizeof tab: %d\n", sizeof tab);
7      printf("sizeof *tab: %d\n", sizeof *tab);
8      printf("sizeof(int): %d\n", sizeof(int));
9
10     for (int i = 0; i < (sizeof tab / sizeof *tab); i++)
11     {
12         printf ("element %d = %d\n", i, tab[i]);
13     }
14     return 0;
15 }
```

Dodatkowo sizeof pozwala zwrócić rozmiar dla konkretnej zmiennej lub dla konkretnego typu, przykład poniżej (kod 4 dostępny w materiałach na stronie):

```
1  #include <stdio.h>
2  int main()
3  {
4      int zmienna = 234;
5      printf("rozmiar zmiennej: %d\n", sizeof(zmienna));
6      printf("rozmiar typu danych int: %d\n", sizeof(int));
7      printf("rozmiar typu danych float: %d\n", sizeof(float));
8      printf("rozmiar typu danych double: %d\n", sizeof(double));
9      printf("rozmiar typu danych char: %d\n", sizeof(char));
10
11     return 0;
12 }
```

Tablice, które mają z góry określony rozmiar są tablicami statycznymi. Poza tablicami statycznymi, możliwe jest użycie tablic dynamicznych. W ich przypadku, rozmiar może się zmienieć w trakcie działania programu. Tablice dynamiczne zostaną omówione w dalszej części zajęć.

2 Tablice dwuwymiarowe

Podobnie jak w przypadku deklaracji tablicy jednowymiarowej, w przypadku tablicy dwuwymiarowej różnież wyróżniamy kilka sposobów deklaracji. Dwa różne sposoby deklaracji tablicy dwuwymiarowej z podaniem jej wartości znajdują się w kodzie poniżej (kod 5 znajduje się w materiałach na stronie):

```
1  #include<stdio.h>
2  int main()
3  {
4      //pierwszy sposob deklaracji macierzy
5      int tab[2][3] = {
6          {1, 2, 3},
7          {4, 5, 6}
```

```
8       };
9
10      //drugi sposob deklaracji macierzy
11      //int tab[2][3] = {1,2,3,4,5,6};
12
13      printf("elementy macierzy:\n");
14      for(int i = 0; i < 2; i++)
15      {
16          for(int j = 0; j < 3; j++)
17          {
18              printf("%d ", tab[i][j]);
19              if(j==2)
20                  printf("\n");
21          }
22      }
23     return 0;
24  }
```

3 Zadania

Zad 1. Napisz program, który pozwala użytkownikowi na wprowadzenie 10 liczb całkowitych z klawiatury, a następnie wyświetla je w odwrotnej kolejności.

Zad 2. Napisz program, który wyznaczy kolejne liczby Fibonacciego, ilość liczb wyznacza użytkownik. w celu wyznaczania elementów ciągu należy wykorzystać tablicę.

Zad 3. Napisz program, który wygeneruje tablicę i wypełni ją losowymi elementami typu int (rozmiar tablicy zadany przez użytkownika). Następnie:

1. Oblicz wartość średnią liczb znajdujących się w tablicy i wypisz ile jest elementów mniejszych od obliczonej wartości średniej.
2. Znajdź element najmniejszy i największy w tablicy.

Zad 4. Napisz program, który prosi użytkownika o podanie wymiaru macierzy (użytkownik określa liczbę wierszy i kolumn), następnie program prosi użytkownika o wprowadzenie wartości dla poszczególnych komórek zadanej macierzy i wyświetla ją.

Zad 5. Napisz program, który pozwala użytkownikowi na wprowadzenie n liczb rzeczywistych do tablicy 10-elementowej, gdzie n jest z zakresu <0,10>. Jeśli użytkownik wprowadzi mniej niż 10 liczb, to dopełnij tablicę zerami (przykładowo jeśli użytkownik poda 4 liczby, to 6 pozostałych liczb przyjmuje wartość 0). Dla tak przygotowanej tablicy oblicz wartość średnią, maksymalną i minimalną.

Zad 6. Napisz program, który wypełni 10-elementową tablicę liczbami pseudolosowymi z przedziału od 0 do 10. Następnie, program losuje ”szczęśliwą liczbę” z tego samego przedziału i sprawdza ile razy wylosowana liczba występuje w tablicy.

Zad 7. Napisz program dodawania macierzy kwadratowych (wykorzystaj tablice dwuwymiarowe).

Zad 8. Napisz program mnożenia macierzy kwadratowych (wykorzystaj tablice dwuwymiarowe).

Zad 9. Napisz program, który dla zadanej przez użytkownika macierzy wyznaczy sumę liczb na głównej przekątnej.

Zad 10. Napisz program, który posortuje tablicę 10 liczb całkowitych (tablica zadana przez użytkownika).