


SPRAWOZDANIE NR 6			
Nazwa ćwiczenia	Wskaźniki i tablice.		 POLITECHNIKA BYDGOSKA Wydział Telekomunikacji, Informatyki i Elektrotechniki
Przedmiot	Podstawy programowania – laboratorium		
Student grupa	Marcin Ogórkiewicz, grupa 7		
Data ćwiczeń	14.12.2022	29.12.2022	Data oddania sprawozdania

Kod oraz wynik działania programu

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 void zadanie1(){
6     int numbers[5], n, i;
7     printf("Zadanie 1\nPodaj 5 liczb całkowitych, a program wyświetli je w odwrotnej kolejności\n");
8     for (i = 0; i < 5; ++i) {
9         scanf(" %d", &n);
10        numbers[i]=n;
11    }
12    for (i = 4; i > 0; --i) {
13        printf(" %d", numbers[i]);
14    }
15    printf(" \n", numbers[0]);
16 }
17
18 void zadanie2() {
19     int matrix[3][3];
20     int num, col = 3, row = 3;
21     printf("Zadanie 2\nPodaj liczby do utworzenia macierzy. Liczby beda dodawane rzadami (najpierw wypelniony zostaje pierwszy rzad trzema liczbami itd.)\n");
22     for (int i = 0; i < row; ++i) {
23         for (int j = 0; j < col; ++j) {
24             scanf(" %d", &num);
25             matrix[i][j] = num;
26         }
27     }
28     printf("Twoja macierz\n");
29     for (int i = 0; i < row; ++i) {
30         for (int j = 0; j < col; ++j) {
```

```
11         printf(" %d ", matrix[i][j]);
12     }
13     printf(" \n");
14 }
15
16 void zadanie3(){
17     int a, b;
18     printf("Zadanie 3\nPodaj pierwsza liczbe\n");
19     scanf(" %d", &a);
20     printf("Podaj druga liczbe\n");
21     scanf(" %d", &b);
22     int wsk_a = &a;
23     int wsk_b = &b;
24     int wsk_c = wsk_a + wsk_b;
25     printf("Wynik dodawania dwoch liczb to: %d\n", wsk_c);
26 }
27
28 void zadanie4(){
29     int tab[10] = { 0: 25, 1: 7, 2: 16, 3: 3, 4: 44, 5: 9, 6: 17, 7: 11, 8: 28, 9: 21};
30     int i, j, temp;
31     printf("Zadanie 4\nNieposortowana tablica: \n");
32     for (i = 0; i < 10; i++)
33     {
34         printf(" %d ", *(tab+i));
35     }
36     printf(" \n");
37     // Sortowanie babelkowe
38     for (i = 0; i < 10; i++)
39     {
```

```
main.c
61     for (j = 0; j < 10-i-1; j++)
62     {
63         if (*(tab+j) > *(tab+j+1))
64         {
65             temp = *(tab+j);
66             *(tab+j) = *(tab+j+1);
67             *(tab+j+1) = temp;
68         }
69     }
70 }
71 printf( format: "Posortowana tablica: \n");
72 for (i = 0; i < 10; i++)
73 {
74     printf( format: "%d ", *(tab+i));
75 }
76 printf( format: "\n");
77 }
78
79 //Funkcja wypełniająca tablice zmiennoprzecinkową liczbami losowymi z zakresu 0 - 1
80 void fill_float_array(float *array, int array_size){
81     srand( seed: time( time: NULL));
82     for(int i = 0; i < array_size; i++){
83         array[i] = (float)rand()/(float)(RAND_MAX);
84     }
85 }
86
87 //Funkcja wypełniająca tablice całkowitą liczbami losowymi z zakresu 0 - 100
88 void fill_int_array(int *array, int array_size){
89     srand( seed: time( time: NULL));
90     for(int i = 0; i < array_size; i++){
```

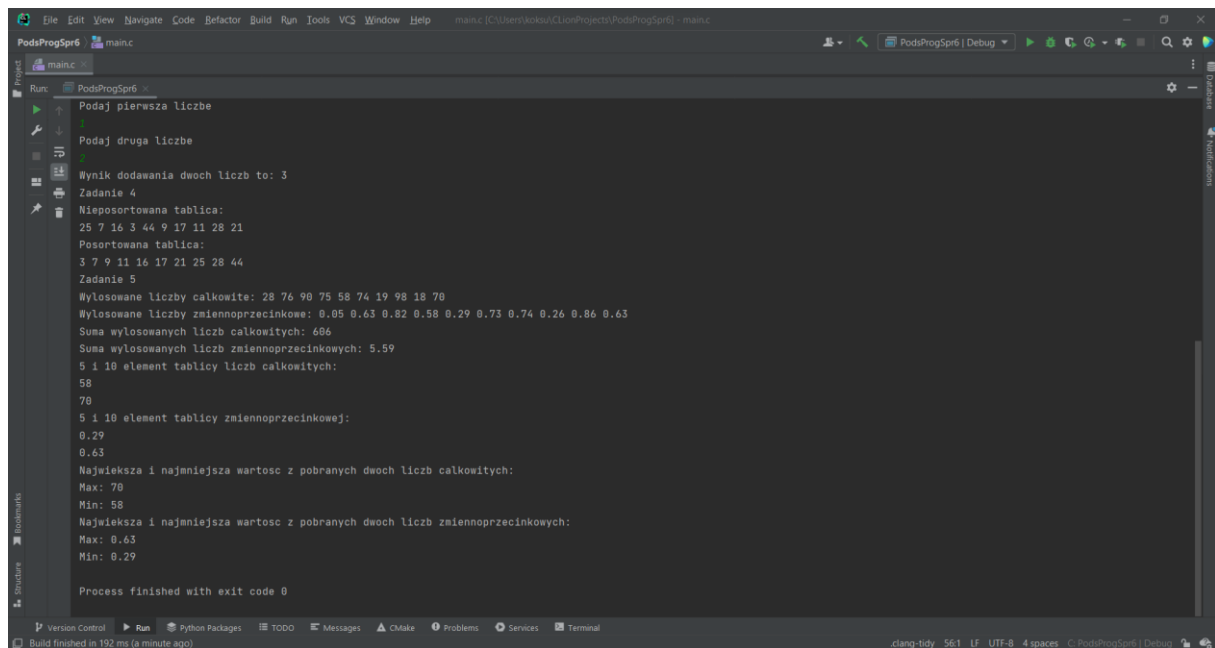
```
PodProgSpr6 | main.c
91     array[i] = rand() % (101 - 0);
92 }
93 }
94
95 //Funkcja sumująca wszystkie liczby z tablicy
96 float sum_float_array(float *array, int array_size){
97     float sum = 0;
98     for(int i = 0; i < array_size; i++){
99         sum += *(array+i);
100     }
101     return sum;
102 }
103
104 int sum_int_array(int *array, int array_size){
105     int sum = 0;
106     for(int i = 0; i < array_size; i++){
107         sum += *(array+i);
108     }
109     return sum;
110 }
111
112 //Funkcja wyświetlająca wszystkie elementy tablicy
113 void print_float_array(float *array, int array_size){
114     for(int i = 0; i < array_size; i++){
115         printf( format: "%.2f ", array[i]);
116     }
117     printf( format: "\n");
118 }
119
120 void print_int_array(int *array, int array_size){
```

```
PodProgSpr6 | main.c
121     for(int i = 0; i < array_size; i++){
122         printf( format: "%d ", *(array+i));
123     }
124     printf( format: "\n");
125 }
126
127 //Funkcja wyświetlająca 5 i 10 element tablicy
128 void print_float_array_elements(float *array){
129     printf( format: "%2f\n", *(array+4));
130     printf( format: "%2f\n", *(array+9));
131 }
132
133 void print_int_array_elements(int *array){
134     printf( format: "%d\n", *(array+4));
135     printf( format: "%d\n", *(array+9));
136 }
137
138
139 //Funkcja znajdująca największą i najmniejszą wartość z dwóch pobranych elementów
140 void min_max_float_elements(float *array){
141     float min, max;
142     if(*(array+4) > *(array+9)){
143         max = *(array+4);
144         min = *(array+9);
145     }
146     else{
147         max = *(array+9);
148         min = *(array+4);
149     }
150     printf( format: "Max: %2f\n", max);
151 }
152
153 zadanie4
Version Control Python Packages TODO CMake Problems Services Terminal
clang-tidy 74:33 LF UTF-8 4 spaces C:PodProgSpr6 | Debug
```

```
PodProgSpr6 | main.c
151     printf( format: "Min: %2f\n", min);
152 }
153 void min_max_int_elements(int *array){
154     int min, max;
155     if(*(array+4) > *(array+9)){
156         max = *(array+4);
157         min = *(array+9);
158     }
159     else{
160         max = *(array+9);
161         min = *(array+4);
162     }
163     printf( format: "Max: %d\n", max);
164     printf( format: "Min: %d\n", min);
165 }
166
167 int main(){
168     zadanie1();
169     zadanie2();
170     zadanie3();
171     zadanie4();
172     int array_size = 10;
173     float float_array[array_size];
174     int int_array[array_size];
175     fill_float_array( array: float_array, array_size);
176     fill_int_array( array: int_array, array_size);
177     printf( format: "Zadanie 5\nWylosowane liczby całkowite: ");
178     print_int_array( array: int_array, array_size);
179     printf( format: "Wylosowane liczby zmiennoprzecinkowe: ");
180     print_float_array( array: float_array, array_size);
181 }
182
183 zadanie4
Version Control Python Packages TODO CMake Problems Services Terminal
clang-tidy 74:33 LF UTF-8 4 spaces C:PodProgSpr6 | Debug
```

```
PodProgSpr6 | main.c
165 }
166
167 int main()
168 {
169     zadanie1();
170     zadanie2();
171     zadanie3();
172     zadanie4();
173     int array_size = 10;
174     float float_array[array_size];
175     int int_array[array_size];
176     fill_float_array(array, float_array, array_size);
177     fill_int_array(array, int_array, array_size);
178     printf(format: "Zadanie 5\nWytlosowane liczby calkowite: ");
179     printf_int_array(array, int_array, array_size);
180     printf(format: "Wytlosowane liczby zmiennoprzecinkowe: ");
181     printf_float_array(array, float_array, array_size);
182     printf(format: "Suma wytlosowanych liczb calkowitych: %d\n", sum_int_array(array, int_array, array_size));
183     printf(format: "Suma wytlosowanych liczb zmiennoprzecinkowych: %.2f\n", sum_float_array(array, float_array, array_size));
184     printf(format: "5 i 10 element tablicy liczb calkowitych: \n");
185     print_int_array_elements(array, int_array);
186     printf(format: "5 i 10 element tablicy zmiennoprzecinkowej: \n");
187     print_float_array_elements(array, float_array);
188     printf(format: "Najwieksza i najmniejsza wartosc z pobranych dwuch liczb calkowitych: \n");
189     min_max_int_elements(array, int_array);
190     printf(format: "Najwieksza i najmniejsza wartosc z pobranych dwuch liczb zmiennoprzecinkowych: \n");
191     min_max_float_elements(array, float_array);
192     return 0;
193 }
```

```
PodProgSpr6 | Run
C:\Users\koku\CLionProjects\PodProgSpr6\cmake-build-debug\PodProgSpr6.exe
Zadanie 1
Podaj 5 liczb calkowitych, a program wyswietli je w odwrotnej kolejnosci
8 7 6 5 4
Zadanie 2
Podaj liczby do utworzenia macierzy. Liczby beda dodawane rzadami(najpierw wypelniony zostaje pierwszy rzad trzema liczbami itd.)
Twoja macierz
1 2 3
4 5 6
7 8 9
Zadanie 3
Podaj pierwsza liczbe
Podaj druga liczbe
```



```
PodProgSpr6
main.c
Run: PodProgSpr6
Podaj pierwsza liczbe
Podaj druga liczbe
Wynik dodawania dwoch liczb to: 3
Zadanie 4
Nieposortowana tablica:
25 7 16 3 44 9 17 11 28 21
Posortowana tablica:
3 7 9 11 16 17 21 25 28 44
Zadanie 5
Wylosowane liczby calkowite: 28 76 98 75 58 74 19 98 18 70
Wylosowane liczby zmiennoprzecinkowe: 0.05 0.63 0.82 0.58 0.29 0.73 0.74 0.26 0.86 0.63
Suma wylosowanych liczb calkowitych: 606
Suma wylosowanych liczb zmiennoprzecinkowych: 5.59
5 i 10 element tablicy liczb calkowitych:
58
70
5 i 10 element tablicy zmiennoprzecinkowej:
0.29
0.63
Najwieksza i najmniejsza wartosc z pobranych dwoch liczb calkowitych:
Max: 70
Min: 58
Najwieksza i najmniejsza wartosc z pobranych dwoch liczb zmiennoprzecinkowych:
Max: 0.63
Min: 0.29
Process finished with exit code 0
```

Zadanie 1

Program inicjuje tablicę liczb całkowitych pięcioelementową oraz zmienne iteracyjną (*i*) i przechowującą aktualny element (*n*). Program prosi użytkownika o wprowadzenie 5 liczb, po czym każdą liczbę przechowuje w tablicy. Następnie, za pomocą pętli *for*, program wypisuje wszystkie liczby od największego do najmniejszego indeksu w tablicy.

Zadanie 2

Program inicjuje tablicę dwuwymiarową liczb całkowitych 3x3, która posłuży do przechowywania macierzy podanej przez użytkownika oraz zmienną *num*, przechowującą wartości podane przez użytkownika. Zdefiniowane zmienne *col* i *row* równe 3 służą do przechowywania i wykorzystywania w pętlach wymiarów tablicy. Następnie, program wypełnia tablicę liczbami podanymi przez użytkownika z wykorzystaniem dwóch pętli. Jedna wypełnia każdą kolumnę tablicy, a druga, nadrzędna, przechodzi pomiędzy wierszami tablicy. Na koniec, w większości tym samym sposobem, macierz zostaje wypisana dla użytkownika.

Zadanie 3

Użytkownik podaje dwie liczby całkowite, które zostają przypisane do zmiennych *a* i *b*. Następnie program tworzy wskaźniki do tych zmiennych, po czym definiuje zmienną *wsk_c*, która dodaje wartości wskazywane przez wskaźniki. Później, program wypisuje wynik dodawania użytkownikowi.

Zadanie 4

Program realizuje sortowanie bąbelkowe na tablicy dziesięcioelementowej *tab*, przy użyciu wskaźników. Najpierw wypisuje tablicę nieposortowaną, po czym dokonuje sortowania. Wybiera elementy od lewej strony zbioru. Jeżeli element wybrany jest większy od elementu po prawej, zamienia je miejscami, do momentu kiedy nie będzie dało się zamienić żadnego elementu miejscami. Na koniec wypisuje tablicę posortowaną.

Zadanie 5

Na początku program losuje liczby całkowite i zmiennoprzecinkowe, po czym wypełnia nimi dwie tabele, według wymagań zadania (funkcje *fill_float_array* oraz *fill_int_array*) oraz wypisuje je (funkcje

`print_float_array` oraz `print_int_array`). Następnie, przy użyciu wskaźników, program sumuje w pętli elementy tablic i wypisuje sumy (funkcje `sum_float_array` oraz `sum_int_array`). Później, również za pomocą wskaźników, program wypisuje 5 i 10 element obydwu tablic (funkcje `print_float_array_elements` oraz `print_int_array_elements`) i wskazuje która z nich jest mniejsza, a która większa (funkcje `min_max_float_elements` oraz `min_max_int_elements`).

Wnioski

Zadania były zrozumiałe i przygotowały mnie do korzystania z tablic oraz wskaźników w języku c.