

Uniwersytet Rzeszowski

PROGRAMOWANIE W C++ – Laboratorium

Nazwisko i imię Paweł Kolano	Laboratorium nr 2 Temat: INSTRUKCJA SWITCH, PĘTLE I TABLICE	Data wykonania: 21.11.2020r.
Grupa laboratoryjna nr 1		Data oddania:
Ocena:		

Zadanie 1.

Przetestuj zamieszczone wyżej fragmenty kodu i sprawdź ich działanie.

Wykonanie:

Zadanie to polegało na przepisaniu fragmentów kodu, które znajdowały się w materiale edukacyjnym mającym nas zapoznać z tematem.

Zadanie 2.

Stwórz pętlę wyświetlającą wartości od 0 do 10 przy pomocy pętli for, while i do..while a następnie napisz 3 pętle wyświetlające liczby od 10 do 0.

Wykonanie:

```
54     for(int i = 0; i <= 10; i++) {
55         cout << i << "\n";
56     }
57
58     int j = 0;
59     cout << "\n";
60
61     while( j <= 10) {
62         cout << j << "\n";
63         j++;
64     }
65
66     cout << "\n";
67     int k = 0;
68
69     do {
70         cout << k << "\n";
71         k++;
72     } while(k <= 10);
```

[Rys 1] – Kod programu, zawierający 6 pętli.

```
9
8
7
6
5
4
3
2
1
0

10
9
8
7
6
5
4
3
2
1
0
Pawels-MacBook-Pro:lab2 lokyiec$
```

[Rys 2] – Część wyniku programu z użyciem pętli.

Program wyświetla liczby od 1 do 10, z użyciem z każdych dostępnych pętli, następnie analogicznie zostają wyświetlone liczby od 10 do 1.

Zadanie 3.

Stwórz tablicę dwuwymiarową do której wprowadzisz wartości tabliczki mnożenia a następnie je wypiszesz.

Wykonanie:

```
105     for (int i = 1; i < 11; i++) {
106         for (int j = 1; j < 11; j++) {
107             tab[i][j] = i * j;
108         }
109         cout << "\n";
110     }
111
112     cout << "Tabliczka mnozenia"<<"\n\n";
113
114     for (int i = 1; i < 11; i++) {
115         for (int j = 1; j < 11; j++) {
116             cout << tab[i][j] << "t";
117         }
118         cout << "\n";
119     }
120 }
```

[Rys 3] - Kod programu dla zadania 3.

```
Tabliczka mnozenia
1   2   3   4   5   6   7   8   9  10
2   4   6   8  10  12  14  16  18  20
3   6   9  12  15  18  21  24  27  30
4   8  12  16  20  24  28  32  36  40
5  10  15  20  25  30  35  40  45  50
6  12  18  24  30  36  42  48  54  60
7  14  21  28  35  42  49  56  63  70
8  16  24  32  40  48  56  64  72  80
9  18  27  36  45  54  63  72  81  90
10 20 30 40 50 60 70 80 90 100
Pawels-MacBook-Pro:lab2 lokyiec$
```

[Rys 4] - Tabliczka mnożenia po wypisaniu.

Program najpierw wypełnia tablicę dwuwymiarową, następnie kolejna pętla wypisuje na ekran.

Zadanie 4.

Wprowadź do tablicy jednowymiarowej 10 kolejnych wyrazów ciągu fibonaciego a następnie je wypisz.

Wykonanie:

```
129     int fibo[10];
130     fibo[0] = 0;
131     fibo[1] = 1;
132     for(int i = 2; i < 10; i++) {
133         fibo[i] = fibo[i-1] + fibo[i-2];
134     }
135
136     for (int j = 0; j < 10; j++) {
137         cout << fibo[j] << "\n";
138     }
```

[Rys 5] - Kod programu obliczający a następnie wypisujący kolejne wyrazy ciągu.

```
0
1
1
2
3
5
8
13
21
34
Pawels-MacBook-Pro:lab2 lokyiec$
```

[Rys 6] - Wynik programu wypisującego kolejne wyrazy ciągu fibbonaciego.

Zadanie to polegało na utworzeniu dwóch pętli, jednej do wypełnienia tablicy, drugiej do wyświetlenia kolejnych wyrazów ciągu fibbonaciego.

Zadanie 5.

Napisz prosty kalkulator wykonujący operacje na dwóch liczbach do jego realizacji wykorzystaj instrukcję switch.

Wykonanie:

```
156 switch (operation) {  
157 |     case '+':  
158 |         cout << x1 << " + " << x2 << " = " << x1 + x2 << "\n";  
159 |         break;  
160 |  
161 |     case '-':  
162 |         cout << x1 << " - " << x2 << " = " << x1 - x2 << "\n";  
163 |         break;  
164 |  
165 |     case '*':  
166 |         cout << x1 << " * " << x2 << " = " << x1 * x2 << "\n";  
167 |         break;  
168 |  
169 |     case '/':  
170 |         cout << x1 << " / " << x2 << " = " << x1 / x2 << "\n";  
171 |         break;  
172 }
```

[Rys 7] - Kod programu kalkulatora.

```
Podaj dwie liczby:  
a = 5  
b = 2  
Jakie dzialanie chcesz wykonac: [+] - suma, [-] - roznica, [*] - iloczyn, [/] - iloraz  
*  
5 * 2 = 10  
PaweLs-MacBook-Pro:lab2 lokyiec$
```

[Rys 8] – Wynik programu obliczający iloczyn.

Wykonanie kalkulatora oparte jest na instrukcji warunkowej switch, dzięki której po wybraniu interesującego nas działania, od razu przechodzimy do odpowiedniego przypadku case.