

Uniwersytet Rzeszowski

Przedmiot:

PROGRAMOWANIE W C++ – Laboratorium

Nazwisko i imię

**Paweł
Kolano**

Data wykonania:

01.11.2020r.

Laboratorium nr 3

Temat:

FUNKCJE REKURENCJA

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 funkcję zwracającą wartość i-tego elementu ciągu Fibonacciego. Numer obliczanego elementu ciągu podaj jako parametr do funkcji.

Wykonanie:

```
25  int fib1(int n) {
26      int a = 0, b = 1, c, i;
27      if ( n == 0) return a;
28
29      for (i = 2; i <= n; i++) {
30          c = a + b;
31          a = b;
32          b = c;
33      }
34      return b;
35  }
```

[Rys 1] - Kod funkcji w programie

```
*****
***** zadanie 2 *****
*****
Podaj nr wyrazu ciągu fibbonaciego: 5
Wyraz: 5 ciagu fibbonaciego to: 5
lokyiec@Pawels-MacBook-Pro lab3 %
```

[Rys 2] - Wynik programu dla 5-tego wyrazu ciągu Fibbonaciego

Zadanie to polegało na utworzeniu algorytmu iteracyjnego obliczającego kolejne wyrazy ciągu Fibonacciego, który ma postać:

$$f_0 = 0$$

$$f_1 = 1$$

$$f_n = f_{n-1} + f_{n-2}$$

Rozwiązanie iteracyjne tego ćwiczenia, polega na utworzeniu pierwszych wyrazów ciągu $a=0$ i $b=1$, oraz pomocniczej zmiennej w przypadku naszego algorytmu zmiennej c .

W przypadku brzegowym algorytmu, jeśli wartość $n=0$, zostanie zwrócona wartość a , jeśli wartość $n>0$, program przejdzie do pętli, która oblicza następny wyraz ciągu przez zsumowanie wyrazów a i b , następnie, nadpisze dotychczasowe wartości $a=b$, $b=c$. Kontynuując iterowanie pętli, aż do uzyskania wartości dla naszego n -tego wyrazu ciągu, zwracając zmienną b .

Zadanie 3.

Napisz prosty kalkulator obliczający sumę, różnicę, iloczyn i iloraz dwóch liczb zmiennoprzecinkowych. Każde z działań ma być realizowane przez odpowiadające działaniom funkcje.

Wykonanie:

```
93 double x1, x2;
94 int operation;
95
96 cout << "Podaj dwie liczby:\n";
97 cout << "a = "; cin >> x1;
98 cout << "b = "; cin >> x2;
99 cout << "Jakie dzialanie chcesz wykonac: 1 - suma, 2 - roznica, 3 - iloczyn, 4 - iloraz\n"; cin >> operation;
100
101 switch (operation) {
102     case 1:
103         cout << x1 << " + " << x2 << " = " << x1 + x2 << "\n";
104         break;
105
106     case 2:
107         cout << x1 << " - " << x2 << " = " << x1 - x2 << "\n";
108         break;
109
110     case 3:
111         cout << x1 << " * " << x2 << " = " << x1 * x2 << "\n";
112         break;
113
114     case 4:
115         cout << x1 << " / " << x2 << " = " << x1 / x2 << "\n";
116         break;
```

[Rys 3] - Kod programu dla zadania 3

```
*****
***** zadanie 3 *****
*****
Podaj dwie liczby:
a = 10
b = 5
Jakie dzialanie chcesz wykonac: 1 - suma, 2 - roznica, 3 - iloczyn, 4 - iloraz
3
10 * 5 = 50
lokyiec@Pawels-MacBook-Pro lab3 %
```

[Rys 4] - Wynik iloczynu dla a=10, b=5

Program pobiera dwie liczby od użytkownika, następnie pyta, jakie działanie ma zostać wykonane, wybór [1-4]. Następnie wczytuje odpowiedź do zmiennej operation, przechodząc do klauzuli switch. Gdzie program wyświetla już oczekiwany rodzaj działania i wynik.

Zadanie 4.

Stwórz funkcję obliczającą wybrany element ciągu Fibonacciego rekurencyjnie.

Wykonanie:

```
37 int fib2(int n) {
38     if (n <= 1) return n;
39     return fib2(n-1) + fib2(n-2);
40 }
```

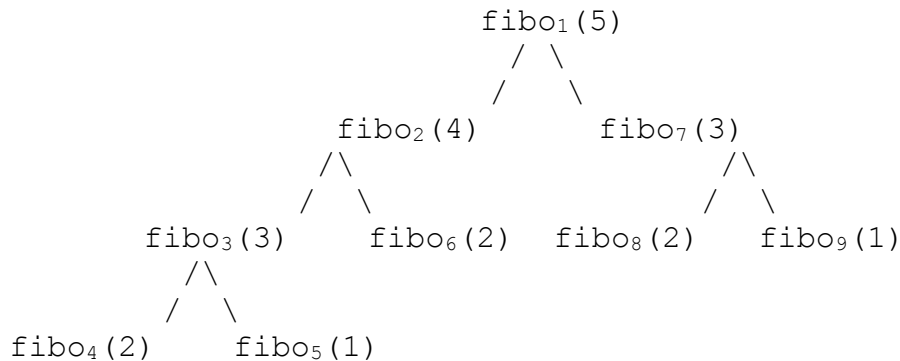
[Rys 5] - Kod programu obliczający rekurencyjnie wyraz ciągu Fibbonacciego

```
*****
***** zadanie 4 *****
*****
Podaj nr wyrazu ciagu fibbonaciego rekurencyjnie: 5
Wyraz: 5 ciagu fibbonaciego to: 5
lokyiec@Pawels-MacBook-Pro lab3 %
```

[Rys 6] - Wynik programu obliczający 5-ty wyraz ciągu

Zadanie to polegało na utworzeniu algorytmu rekurencyjnego obliczającego kolejne wyrazy ciągu Fibonacciego.

Algorytm ten wykorzystuje rekurencję, czyli wywoływanie funkcji przez samą siebie, w razie przypadku brzegowego, klauzura *if*($n \leq 1$), *funkcja zwróci nam pierwszy wyraz ciągu* $f_1=0$. Jeśli n będzie większe, nastąpi wywołanie $\text{fib2}(n-1) + \text{fib2}(n-2)$. Przebieg rekurencji przedstawia poniższe drzewo:



Zadanie 5.

Stwórz funkcję obliczającą silnię rekurencyjnie.

Wykonanie:

```

42  int factorial(int n) {
43      if (n > 1) return n * factorial(n - 1);
44      else return 1;
45  }
  
```

[Rys 7] - Kod programu obliczający rekurencyjnie silnie liczby naturalnej

```

*****
***** zadanie 5 *****
*****

Podaj liczbę naturalną: 5

Silnia rekurencyjnie dla 5! = 120

lokyiec@Pawels-MacBook-Pro lab3 %
  
```

[Rys 8] - Wynik programy obliczające silnię dla liczby 5

Zadanie jest bardzo podobne do zadania 4. Polega na zdefiniowaniu warunków kiedy funkcją zakończy wykonywanie i zostanie zwrócony wynik z działaniem silni.