

# Laboratorium 4

Szymon Kałuża

22.03.2022

## Zadanie 1

```
F(1):  
F(1 - 2)  
    return;  
F(1 - 3)  
    return;
```

Sekwencja liczb: 1,1

```
F(2):  
F(2 - 2)  
    return;  
F(2 - 3)  
    return;
```

Sekwencja liczb: 2,2

```
F(3):  
F(3 - 2)  
    F(1 - 2)  
        return;  
    F(1 - 3)  
        return;
```

```
F(3 - 3)  
    return;
```

Sekwencja liczb: 3,1,1,3

```
F(4):  
F(4 - 2)  
    F(2 - 2)  
        return;  
    F(2 - 3)  
        return;
```

```
F(4 - 3)  
    F(1 - 2)  
        return;  
    F(1 - 3)  
        return;
```

Sekwencja liczb: 4,2,2,1,1,4

```
F(5):  
F(5 - 2)  
    F(3 - 2)  
        F(1 - 2)  
            return;  
        F(1 - 3)  
            return;
```

```
F(3 - 3)  
    return;
```

```
F(5 - 3)  
    F(2 - 2)  
        return;  
    F(2 - 3)
```

```
        return;
Sekwencja liczb: 5,3,1,1,3,2,2,5
```

```
F(6):
print(6);
F(6 - 2)
    print(4)
    F(4 - 2)
        print(2)
        F(2 - 2)
            return;
        F(2 - 3)
            return;
        print(2)
    F(4 - 3)
        print(1)
        F(1 - 2)
            return;
        F(1 - 3)
            return;
        print(1)
    print(4)
F(6 - 3)
    print(3)
    F(3 - 2)
        print(1)
        F(1 - 2)
            return;
        F(1 - 3)
            return;
        print(1)
    F(3 - 3)
        return;
    print(3)
print(6);
Sekwencja liczb: 6,4,2,2,1,1,4,3,1,1,3,6
```

## Zadanei 2

```
F(1):
F(1 - 3)
    return
F(1 - 2)
    return
Sekwencja liczb: 1,1

F(2):
F(2 - 3)
    return;
F(2 - 2)
    return;
Sekwencja liczb: 2,2

F(3):
F(3 - 3)
    return;
F(3 - 2)
    F(1 - 3)
        return;
    F(1 - 2)
        return;
Sekwencja liczb: 3,1,1,3
```

```

F(4):
F(4 - 3)
    F(1 - 3)
        return;
    F(1 - 2)
        return;
F(4 - 2)
    F(2 - 3)
        return;
    F(2 - 2)
        return;
Sekwencja liczb: 1,1,4,2,2,4

```

```

F(5)
F(5 - 3)
    F(2 - 3)
        return;
    F(2 - 2)
        return;
F(5 - 2)
    F(3 - 3)
        return;
    F(3 - 2)
        F(1 - 3)
            return;
        F(1 - 2)
            return;
Sekwencja liczb: 2,2,5,3,1,1,3,5

```

```

F(6):
F(6 - 3)
    F(3 - 3)
        return;
    print(3)
    F(3 - 2)
        F(1 - 3)
            return;
        print(1)
        F(1 - 2)
            return;
        print(1)
    print(3)
print(6);
F(6 - 2)
    F(4 - 3)
        F(1 - 3)
            return;
        print(1)
        F(1 - 2)
            return;
        print(1)
    print(4)
    F(4 - 2)
        F(2 - 3)
            return;
        print(2)
        F(2 - 2)
            return;
        print(2);
    print(4);
print(6)
Sekwencja liczb: 3,1,1,3,6,4,2,2,1,1,4,6

```

## Zadanie 3

```
F(1):
F(1 - 3)
    return
F(1 - 2)
    return
Sekwencja liczb: 1,1

F(2):
F(2 - 3)
    return;
F(2 - 2)
    return;
Sekwencja liczb: 2,2

F(3):
F(3 - 3)
    return;
F(3 - 2)
    F(1 - 3)
        return;
    F(1 - 2)
        return;
Sekwencja liczb: 1,1,3,3

F(4):
F(4 - 3)
    F(1 - 3)
        return;
    F(1 - 2)
        return;
F(4 - 2)
    F(2 - 3)
        return;
    F(2 - 2)
        return;
Sekwencja liczb: 1,1,2,2,4,4

F(5)
F(5 - 3)
    F(2 - 3)
        return;
    F(2 - 2)
        return;
F(5 - 2)
    F(3 - 3)
        return;
    F(3 - 2)
        F(1 - 3)
            return;
        F(1 - 2)
            return;
Sekwencja liczb: 2,2,3,1,1,3,5,5

F(6):
F(6 - 3)
    F(3 - 3)
        return;
    F(3 - 2)
        F(1 - 3)
            return;
        F(1 - 2)
            return;
```

```
        print(1)
        print(1)
    print(3)
    print(3)
F(6 - 2)
print(6)
print(6)
Sekwencja liczb: 3,1,1,3,4,2,2,1,1,4,6,6
```

## Zadanie 4

Mnożenie

## Zadanie 5

Mnożenie

## Zadanie 6

Potęgowanie

## Zadanie 7

Ciąg fibonacciego

## Zadanie 8

```
1 void IntToBinary(int number)
2 {
3     if(number == 0) return;
4
5     IntToBinary(number/2);
6     printf("%d", number%2);
7 }
```

## Zadanie 9

```
1 void piramida(int level)
2 {
3     if(level == 0) return;
4     for(int j = 0; j < level; j++)
5     {
6         printf("*");
7     }
8     printf("\n");
9     piramida(level - 1);
10
11 }
```

## Zadanie 10

```
1 void piramida2(int level)
2 {
3     if(level == 0) return;
4     piramida2(level - 1);
5     for(int j = 0; j < level; j++)
6     {
7         printf("*");
8     }
9     printf("\n");
10
11 }
```

## Zadanie 11

```
1 float Ex11A(float n){
2     if(n == 0) return 0;
3     if(n == 1) return 1;
4     return Ex11A(n - 1) + (1/n)*(1/n);
5 }
6
7 int Ex11B(int n){
8     if(n == 0) return 0;
9     return Ex11B(n - 1) + n;
10 }
11
12 int Ex11C(int n){
13     if(n == 0) return 0;
14     return Ex11C(n - 1) + n * 2;
15 }
16
17 int Ex11D(int n){
18     if(n == 0) return 0;
19     return Ex11D(n - 1) + (n *(n + 1)*((2 * n) + 1))/6;
20 }
```

## Zadanie 12

```
1 double zlotyPodzial(int n){
2     if(n == 0) return 1;
3     return 1 + 1/zlotyPodzial(n-1);
4 }
```

## Zadanie 13

```
1 double F13(int n){
2     if(n == 1) return -1;
3     return -F13(n - 1) * n - 3;
4 }
```

## Zadanie 14

```
1 long pascalTriangle(int n, int i){
2     if(n == i || i == 0) return 1;
3     return pascalTriangle(n - 1, i) + pascalTriangle(n - 1, i - 1);
4 }
```

## Zadanie 15

Kod programu:

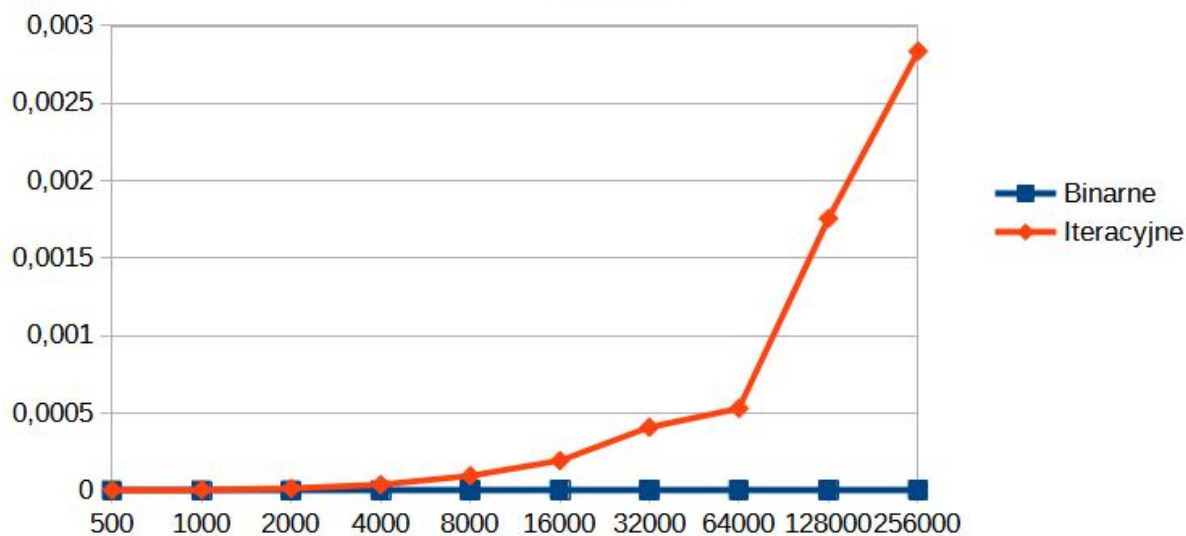
```
1 int binaryNWD(int p, int q){
2     if(q == 0) return p;
3     if(p == 0) return q;
4     if((p % 2 == 0) && (q % 2 == 0)) return 2 * binaryNWD(p/2, q/2);
5     if((p % 2 == 0)&&(q % 2 != 0)) return binaryNWD(p/2, q);
6     if((p % 2 != 0)&&(q % 2 == 0)) return binaryNWD(p, q/2);
7     if((p % 2 != 0)&&(q % 2 != 0)&&(p >= q)) return binaryNWD((p-q)/2, q);
8     if((p % 2 != 0)&&(q % 2 != 0)&&(p < q)) return binaryNWD(p, (q - p)/2);
9 }
```

Kod programu udostępnionego na zajęciach:

```
1 long NWD (int n, int k){
2     if (k == 0) return n;
3     return NWD(k, n % k);
4 }
```

Porównanie:

## Porównanie szybkości działania wyszukiwania binarnego z iteracyjnym



[bhp]

Figure 1: Porównanie drugie

## Zadania 16

```

1 void reverse(int* tab, int n, int k){
2     int rightIndex = n - 1;
3     int leftIndex = k - n;
4     //printf("left index: %d right Index: %d", leftIndex, rightIndex);
5     if(leftIndex > rightIndex) return;
6     int temp = tab[leftIndex];
7     tab[leftIndex] = tab[rightIndex];
8     tab[rightIndex] = temp;
9     reverse(tab, n-1, k);
10 }

```

## Zadanie 17

Kod programu wyszukiwania binarnego:

```

1 int binarySearch(int* tab, int left, int right, int x)
2 {
3     if (right >= left)
4     {
5         int mid = left + (right - left)/2;
6         if (tab[mid] == x) return mid;
7         if (tab[mid] > x) return binarySearch(tab, left, mid-1, x);
8         return binarySearch(tab, mid+1, right, x);
9     }
10    return -1;
11 }

```

Kod programu wyszukiwania iteracyjnego:

```

1 int iterativeSearch(int* tab, int n, int x){
2     if(tab[n]==x) return n;
3     else return iterativeSearch(tab, n - 1, x);
4     return -1;
5 }

```

Porównanie: