

Увод в алгоритмите и програмирането

Домашна работа №3

Никола Николов, F106501

Задача 1: *Задача 1. Напишете формулата на Гаус за сумата на първите n естествени числа. Формулата е дадена в презентацията към тази тема. Под нея напишете проверка - дали формулата е вярна за сумата на първите k естествени числа, където k е последната цифра от факултетния ви номер. Ако номерът ви завършва с нула, приемайте $k=6$.*

Решение:

$$\text{Формула на Гаус за сумата на първите } n \in \mathbb{N}: \sum_{i=1}^n i = \frac{n(n+1)}{2} \quad (*)$$

Нека $k = 6$.

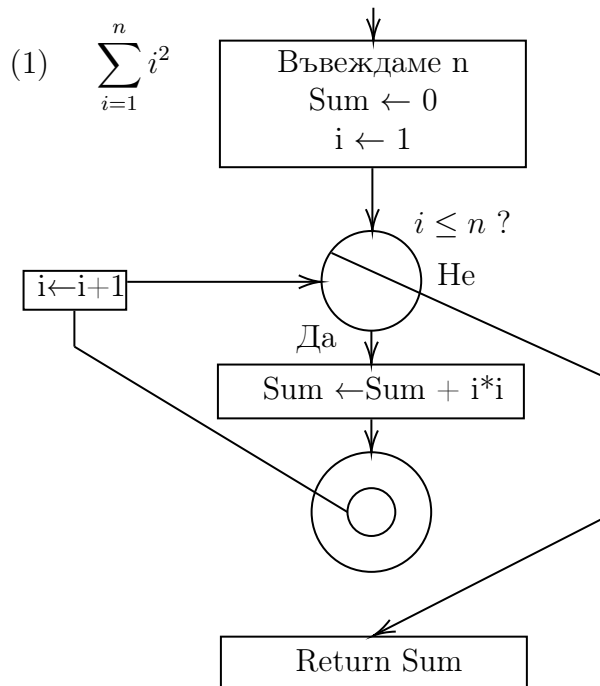
$$\sum_{i=1}^6 i = 1 + 2 + 3 + 4 + 5 + 6 = 21$$

$$\text{Проверяваме по формулата на Гаус: } \frac{n(n+1)}{2} = \frac{6 \cdot 7}{2} = 21$$

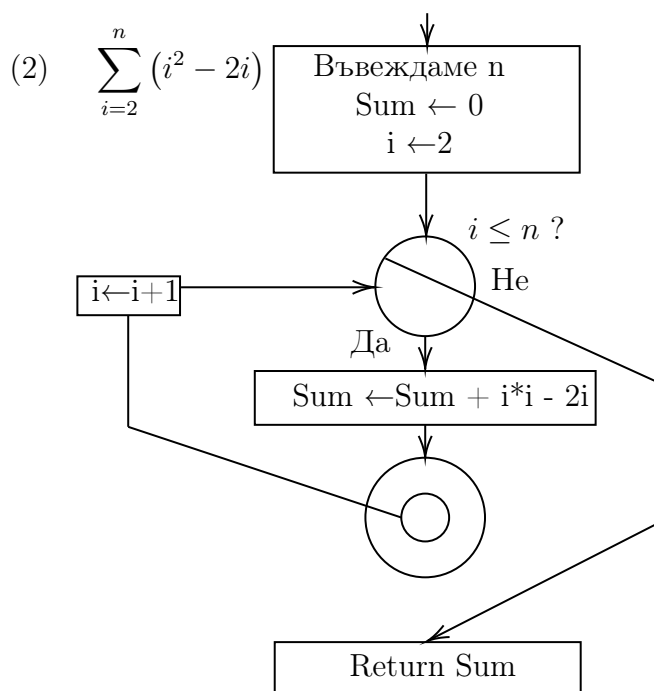
\Rightarrow формула (*) е изпълнена за $k = 6$.

Задача 2: *Съставете трите схеми на управление, успоредно на тях - програмния текст на познат за вас език и пуснете на машина една от програмите за натрупване на следните три суми:*

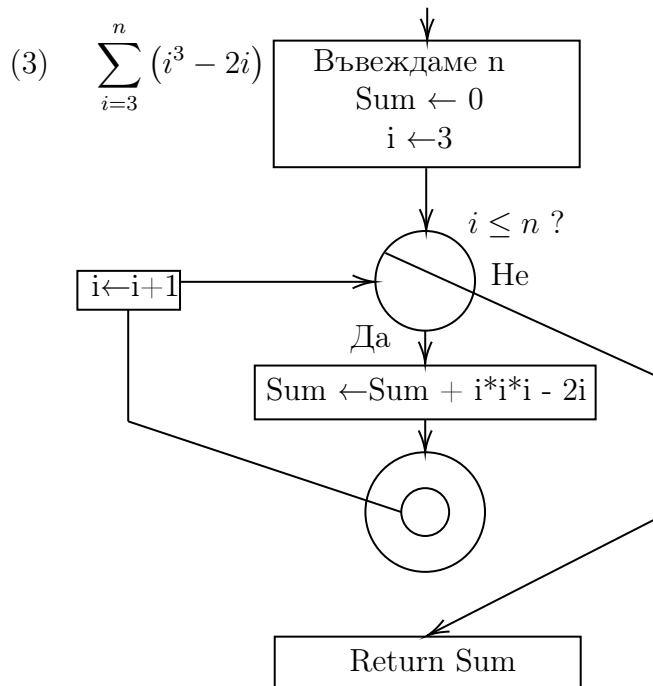
$$(1) \sum_{i=1}^n i^2, \quad (2) \sum_{i=2}^n (i^2 - 2i), \quad (3) \sum_{i=3}^n (i^3 - 2i)$$



```
//C++ code
int sum1(int n)
{
    int sum=0;
    for(int i=1;i<=n;i++){
        sum+= i*i;
    }
    return sum;
}
```



```
//C++ code
int sum2(int n)
{
    int sum=0;
    for(int i=2;i<=n;i++){
        sum+= i*i-2*i;
    }
    return sum;
}
```



```
//C++ code
int sum3(int n)
{
    int sum=0;
    for(int i=3;i<=n;i++){
        sum+= i*i*i-2*i;
    }
    return sum;
}
```

Изход при подадено $n = 101$ на функцията "sum2":

```
8 int main()
9 {
10     int n = 101;
11     std::cout << sum2(n) << '\n';
}
```

```
nn@NNPC: ~/Programming/NBU/Algoritmi/Week3
nn@NNPC:~/Programming/NBU/Algoritmi/Week3$ g++ sum.cpp -o sum
nn@NNPC:~/Programming/NBU/Algoritmi/Week3$ ./sum
338250
```

Задача 3: Като се базирате на разсъждения, подобни на тези в алгоритъма за натрупване на сума, съставете програма за пресмятане на произведението факториел на числото n , задавано от клавиатура. Не се приемат решения, базирани на копирана отнякъде рекурсия. Искане за цикъл.

Решение:

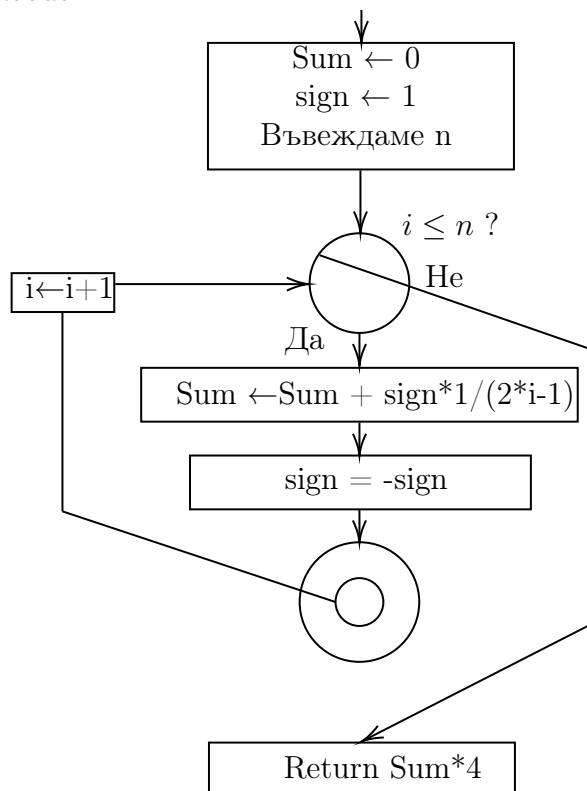
Програма на C++ работеща коректно за $n \in \{0, \dots, 20\}$:

```
unsigned long long int factorial(int n)
{
    unsigned long long int sum = 1;
    for(int i=1; i<=n; i++){
        sum *= i;
    }
    return sum;
}
```

Задача 4: Съставете схемата на управление (има я в лекцията) и успоредно на нея – програмата за пресмятане на с алтернативен ред. Съставете програмата по ДВА начина:

а) Сумата се пресмята 10 000 пъти, по брояч. На всеки 500 преминавания през цикъла, изчислената стойност за да се извежда на екран до 15 знак след десетичната запетая.

б) Сумата се пресмята ДОТОГАВА, ДОКАТО разликата между две стойности, получени последователно, се получи по-малка от 0.0000001. Програмата да извежда на екран след колка преминавания през цикъла се постига това условие.



```
#include <iostream>
long double leibnizSeries(int n);

int main()
{
    std::cout.precision(16);
    leibnizSeries(10000);

    return 0;
}

long double leibnizSeries(int n)
{
    long double sum=0;
    int sign = 1;
    for(int i=1; i<=n; i++){
        sum += sign*1.0/(2*i-1);
        sign = -sign;
        if(i%500==0)
            std::cout << 4*sum << '\n';
    }
    return 4*sum;
}
```

Исход от кода на зад. 4a):

```
nn@NNPC:~/Programming/NBU/Algoritmi/Week3/HW3$ ./zad4
3.139592655589783
3.140592653839793
3.140925986997201
3.141092653621043
3.141192653605793
3.141259320265719
3.14130693930991
3.1413426535937
3.141370431370315
3.141392653591793
3.141410835409478
3.141425986924284
3.141438807436858
3.141449796447665
3.141459320257053
3.141467653590282
3.141475006531377
3.141481542479025
3.14148739043219
3.141492653590043
```

Задача 4.б):

```
//C++ code
#include <iostream>
#include <cmath>

long double leibnizSeries();

int main()
{
    std::cout.precision(16);

    leibnizSeries();
    return 0;
}

long double leibnizSeries()
{
    long double sum=0;
    int sign = 1;
    long double tmp=0;
    int cnt=1;
    long double diff = 0;
    do{
        tmp = sum;
```

```

        sum += sign*1.0/(2*cnt-1);
        sign = -sign;
        cnt++;
    }while( fabs(sum-tmp) > 0.0000001 );

    std::cout << "Approximation after " << cnt-1 << " loops:" << '\n' << 4*sum;

    return 4*sum;
}

```

Исход от кода на зад. 4б):

```

nn@NNPC:~/Programming/NBU/Algoritmi/Week3/HW3$ ./zad4b
Approximation after 5000001 loops:
3.141592853589753

```