МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Физико-технический факультет

Кафедра информатики и вычислительной техники (ИВТ)

Лабораторная работа №03

**Программирование с использованием циклов**

Выполнил студент 585 гр.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.М. Губченко

Проверил: к.т.н,, доцент каф. ВТиЭ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ю.Г. Скурыдин

Лабораторная работа защищена

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019 г.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Барнаул 2019

1. **Формулировка задачи**

Ввести с клавиатуры последовательность из k действительных чисел и число n, не равное ни одному из чисел последовательности. Найти ближайшее к n число последовательности (его порядковый номер и значение)

1. **Постановка задачи**

На вход поступает целое положительное число k, определяющее размер числовой последовательности, произвольное число n и последовательность из k вещественных чисел. Программа находит среди чисел введенной последовательности ближайшее к k число. На выход программы поступает целое положительное число, определяющее порядковый номер ближайшего к n числа и вещественное число, определяющее его значение.

1. **Математическая модель**

Для построения математической модели рассмотрим пример. Инициализируется переменная number с значением 0. Программа получает на вход число k=4, n=4, x1=5, x2=6, x3=7, x4=8, в процессе ввода чисел последовательности, программа выполняет увеличение переменной number на 1 вычитание из числа последовательности число n, определяется остаток, если он меньше значения переменной z, то переменной z присваивается численное значение данного остатка, переменной out\_number присваивается значение number, переменной out\_x присваивается значение переменной x, иначе продолжается ввод значений переменной x. В конце работы программы, мы получим порядковый номер 1 и значение числа равное 5.

1. **Описание алгоритма**

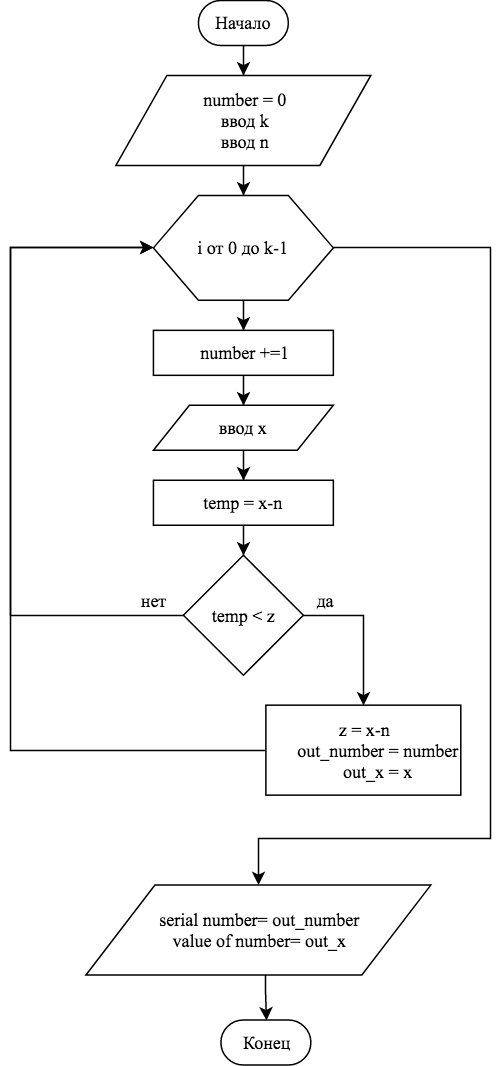
**Начало алгоритма**

* 1. Присвоение переменной number значения 0
  2. Присвоение переменной z значения 3.4е38
  3. ввод числа k
  4. ввод числа n
  5. присвоение переменной i значения 0
  6. Если i < k, переход к пункту 4.7, иначе переход к 4.13
  7. ввод числа x
  8. присвоение переменной number значения number +1
  9. присвоение переменной temp значения x – n
  10. присвоение переменной i значения i+1
  11. если temp < z, то присвоение переменной z значения x – n, присвоение переменной out\_number значения number, присвоение переменной out\_x значения x, иначе переход к пункту 4.12
  12. переход к пункту 4.6
  13. вывод serial number= и out\_number
  14. вывод value of number= и out\_x

**Конец алгоритма**

1. **Проект программы с нахождением ближайшего к n числа из последовательности из k чисел**

Проект программы с определением замкнутых программных единиц и необходимых структур данных. В данном случае программа должна состоять из одиннадцати частей: 3 блока ввода, цикла с параметром (содержит блок определения переменной i, 2 блока проверки условия, два блока арифметических действий и блок ввода), 1 блок проверки условия, 5 блоков арифметических вычислений и 2 блока вывода. В качестве структуры данных могут быть введены переменные для хранения исходного числа, переменной x, результата.

1. **Опорный граф (блок-схема) алгоритма**
2. **Текст программы:**

**на языке С++**

#include <iostream>

#include <cstdlib>

#include <math.h>

using namespace std;

int main (int argc, char const \*argv[]) {

int k, number, out\_number;

float x, z, n, temp, out\_x;

z = 3.4e38;

number = 0;

cout << "k=";

cin >> k;

cout << "n=";

cin >> n;

for (int i=0; i<k; i++) {

number += 1;

cout << "x" << number << "=";

cin >> x;

temp = x - n;

if (abs(temp) < z) {

z = abs(x-n);

out\_number = number;

out\_x = x;

}

}

cout << "serial number= " <<out\_number<<'\n';

cout << "value of number= " << out\_x<<'\n';

return 0;

}

**на языке Pascal**

Program lab\_03;

var number, out\_number: real;

var k, out\_x, z, n, x, i, temp, k: integer;

Begin

z:= 32767;

number:= 0;

write('k=');

readln(k);

Write('n=');

ReadLn(n);

for i := 0 to k-1 do begin

number:= number + 1;

Write('x=');

ReadLn(x);

temp:= x - z;

if (abs(temp) < z) then begin

z:= abs(x-n);

out\_number:= number;

out\_x:= x;

end;

end;

WriteLn('serial number=', out\_number);

WriteLn('value of number=', out\_x);

readln;

end.

1. **Проверка работоспособности (тестирование) программы**

Для проверки работоспособности программы, переменной k присвоим значение 4, переменной n значение 5, переменным x1, x2, x3 и x4 присвоим значения 1, 2, 3 и 4 соответственно. Т.к. значение x, ближайшее к n, равно 4, то программа выведет значение переменной x4 и его порядковый номер(4).

1. **Сравнительный анализ и оценка эффективности работы программ на разных языках программирования**

Ввиду простоты программы она одинаково эффективно выполняется для обоих языков программирования.

1. **Формулировка задачи**

Пусть Дано натуральное n. Найти

1. **Постановка задачи**

На вход поступает натуральное число n. Программа высчитывает сумму первых n членов бесконечного числового ряда, получаемого с использованием рекуррентных формул. На выход поступает вещественное число в диапазоне от -∞ до +∞

1. **Математическая модель**

Для построения математической модели рассмотрим пример. Инициализируются переменные x1 и y1 равные 1. Программа получает на вход натуральное число n=4. После этого программа вычисляет сумму первых n членов последовательности, каждый член которой задаётся в виде , где i- это порядковый номер. xi задаётся формулой xi=0.3\*xi-1, а yi задаётся формулой yi=xi-1+yi-1. При n=1 x1=1, y1=1, sum=0.5. При n=2 x2=0.3, y2=2, sum=0.6. При n=3 x3=0.09, y3=2.3, sum= 0.627273. При n=4 x4=0.027, y4=2.39, sum=0.635237. На выход поступает действительное число 0.635237

1. **Описание алгоритма**

**Начало алгоритма**

4.1 Присвоение переменной k значения 1

4.2 Присвоение переменной x1 значения 1

4.3 Присвоение переменной y1 значения 1

4.4 Присвоение переменной sum значения 0

4.5 Ввод числа n

4.6 Присвоение переменной xi значения x1

4.7 Присвоение переменной yi значения y1

4.8 Присвоение переменной i значения 0

4.9 Проверка условия i<n. Если не выполняется, то переход к пункту 4.16

4.10 Проверка условия k>1. Если не выполняется то переход к пункту 4.14

4.11 Присвоение переменной yi значения xi + yi

4.12 Присвоение переменной xi значения 0.3 \* xi

4.13 Присвоение переменной k значения k+1

4.14 Присвоение переменной sum значения sum+xi/(1+abs(yi))

4.15 присвоение переменной i значения i+1 Переход к пункту 4.9

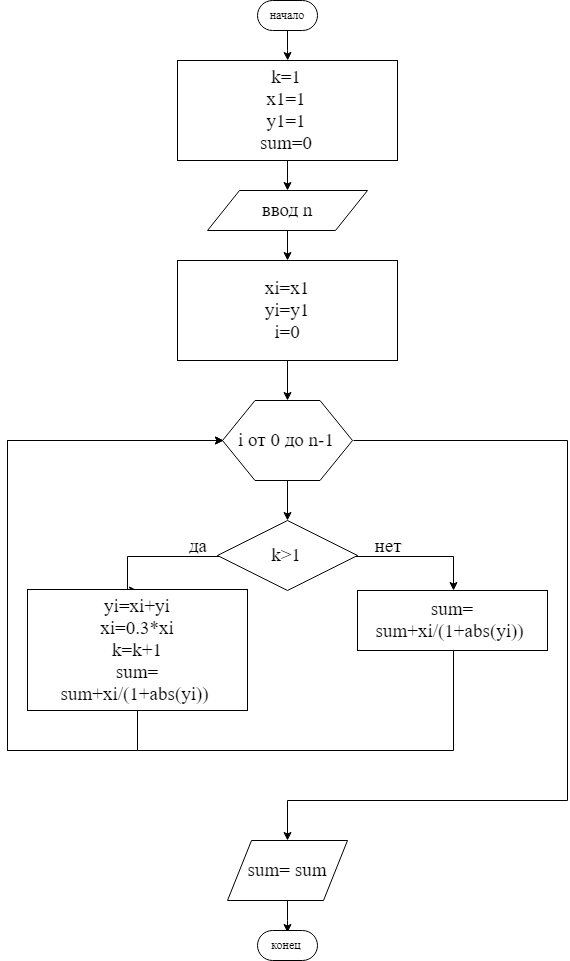
4.16 Вывод “sum=” и sum

**Конец алгоритма**

1. **Проект программы с нахождением суммы первых n членов последовательности**

Проект программы с определением замкнутых программных единиц и необходимых структур данных. В данном случае программа должна состоять из одиннадцати частей: 1 блок ввода, цикла с параметром (содержит блок определения переменной i, 2 блока проверки условия, 6 блоков арифметических действий) и блок вывода. В качестве структуры данных могут быть введены переменные для хранения исходного числа, переменной x, результата.

1. **Опорный граф (блок-схема) алгоритма**



1. **Текст программы:**

**на языке С++**

#include <iostream>

#include <cstdlib>

#include <math.h>

using namespace std;

int main (int argc, char const \*argv[]) {

int x1, y1, n, k;

float sum, xi, yi;

k=1;

x1=1;

y1=1;

sum=0;

xi = x1;

yi = y1;

cout << "n=";

cin >> n;

for (int i=0; i<n; i++) {

if (k>1) {

yi= xi + yi;

xi= 0.3 \* xi;

k+=1;

sum+= xi/(1+abs(yi));

} else {

sum+= xi/(1+abs(yi));

k+=1;

}

}

cout << "sum= " <<sum<<'\n';

return 0;

}

**на языке Pascal**

Program lab\_03;

var sum, xi, yi: real;

var x1, y1, n, k, i: integer;

Begin

k:=1;

x1:=1;

y1:=1;

sum:=0;

xi:=x1;

yi:=y1;

Write('n=');

ReadLn(n);

for i := 0 to n-1 do begin

if (k>1) then

begin

yi:= xi + yi;

xi:= 0.3 \* xi;

k:= k+1;

sum:= sum + xi/(1+abs(yi));

end else

begin

sum:= sum + xi/(1+abs(yi));

k:= k+1;

end;

end;

WriteLn('sum=', sum);

end.

1. **Проверка работоспособности (тестирование) программы**

Для проверки работоспособности программы, переменной n присвоим значение 1. Тогда xi будет равно 1, а yi будет равно 1. Тогда xi/(1+|yi|) будет равно 0.5

1. **Сравнительный анализ и оценка эффективности работы программ на разных языках программирования**

Ввиду простоты программы она одинаково эффективно выполняется для обоих языков программирования.