МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Институт цифровых технологий, электроники и физики

Кафедра вычислительной техники и электроники (ВТиЭ)

Лабораторная работа № 3

**Программирование с использованием циклов**

Выполнил студент 595 гр.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.В. Осипенко

Проверил: к.т.н,, доцент каф. ВТиЭ

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

Лабораторная работа защищена

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

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

Барнаул 2020

1. **Формулировка задачи**

Дано 20 вещественных чисел. Вычислить разность между максимальным и минимальным числом.

1. **Постановка задачи**

На вход программы поступают 20 вещественных чисел в диапазоне +/- 1.7e308, необходимо сравнить их друг с другом и выявить наибольшее и наименьшее число, затем найти разность между максимальным и минимальным числом. На выход программы поступают три вещественных значения чисел min, max и результат выражения: max - min.

1. **Математическая модель**

Нужно сравнить 20 вещественных чисел и найти среди них наибольшее и наименьшее. Для этого берем 1ое число и делаем его и max и min, затем берем 2ое и сравниваем с max и min, если оно больше max или меньше min, то переопределяем его max или min, иначе переходим к следующему числу, с которым делаем тоже самое и т.д. Например: num1 = 23.4, max = 23.4, min = 23.4; num2 = 30.0, max = 30.0, min = 23.4; … в итоге например получим: max=548.9, min = -12.5, max – min = 561,4 .

1. **Описание алгоритма**

Начало

* 1. Вводим число num
  2. Присвоить max = num, min=num, i = 1
  3. Проверка условия: i <= 19

Если условие не выполняется, то вывод значений max и min, разность значений min и max, завершение работы программы

* 1. Вводим число num
  2. Проверка условия: (num > max)

Если условие не выполняется, то переход к пункту 4.9

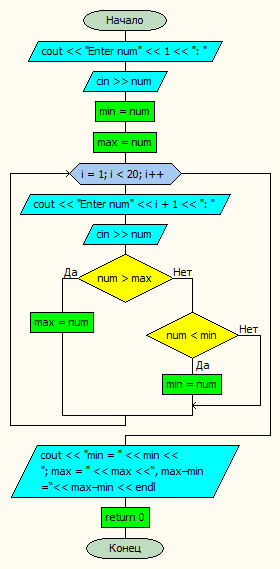
* 1. Присвоить max значение num, перейти к пункту 4.9
  2. Проверка условия: (num < min)

Если условие не выполняется, то переход к пункту 4.9

* 1. Присвоить min значение num
  2. Увеличить i на 1, перейти к пункту 4.3

Конец

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

****

1. **Проект программы с определением замкнутых программных единиц и необходимых структур данных**

Программа состоит из одного блока цикла, включающего себя два блока условий, двух блоков ввода и трех блоков вывода. Структура данных – нет.

1. **Текст программы:**

**на языке С++**

#include <iostream>

using namespace std;

int main()

{

double num,min,max;

cout << "Enter num"<< 1 << ": "; cin >> num;

min = num; max = num;

for(int i = 1; i <= 19; i++){

//Ввод num

cout << "Enter num"<< i+1 << ": "; cin >> num;

// Проверка на min/max

if(num > max) max = num;

else if(num < min) min = num;

}

cout << "min = " << min << "; max = " << max << “, max – min = “ << max – min<< endl;

return 0;

}

**на языке Pascal**

var

num,max,min:real;

i: integer;

begin

writeln('Enter num', 1); readln(num);

min := num; max := num;

for i := 1 to 19 do

begin

//Ввод num

writeln('Enter num',i+1); readln(num);

// Проверка на min/max

if num > max then max := num

else if num < min then min := num;

end;

writeln('max = ',max,' min = ',min, “, max – min = “ ,max – min);

end.

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

Введем : 36 -11 29 84 51 73 59 5 13 34 97 66 46 92 22 7 44 50 76 30 19 52 91 40 32 18 79 88 67 20 9 81 35 69 83 77 23 96 25 17 3 37 56 47 31 43 75 55 10 78, результат: max =97.0 , min =-11.0, max – min = 108 .

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

С++: 0.02 сек, pascal: 1.1 сек. Данная программа на C++ выполняется в среднем 55 раз быстрее чем на pascal

1. **Формулировка задачи**

Для числового ряда, общий член которого имеет вид , найти сумму тех членов ряда, для которых , где значение 0< ε < 1 вводится с клавиатуры

1. **Постановка задачи**

На вход программы поступает одно вещественное число в диапазоне от 0 до 1. Программа выполняет расчет значения суммы элементов числового ряда с заданной точностью. На выход программы поступает одно вещественное число,

1. **Математическая модель**

Дан числовой ряд {a\_1, a\_2, a\_3, a\_4, … , a\_n-1, a\_n : |a\_n|>=eps}, каждый член которого задан формулой , нужно найти сумму n-ых элементов. Пример: eps = 0.01, a\_1=0.25 >= eps, a\_2=0.0357 >= eps, a\_3=0.014>= esp, a\_4 = 0.008 < eps(не удов условию, значит мы складываем только первые 3 члена ряда); sum = 0.25+0.0357+0.014 = 0.3.

1. **Описание алгоритма**

Начало

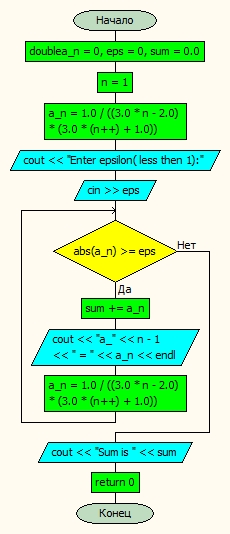
* 1. Вводим число eps
  2. Присвоить n = 1, sum = 0.0, a\_n = 1.0/((3.0\*n-2.0)\*(3.0\*(n++)+1.0))
  3. Ввод числа eps
  4. Проверка условия: abs(a\_n) >= eps

Если условие не выполняется вывести sum, завершить работу программы

* 1. Присвоить sum = sum + a\_n
  2. Вывод значения a\_n
  3. Присвоить a\_n = 1.0/((3.0\*n-2.0)\*(3.0\*(n++)+1.0)), перейти к пункту 4.4

Конец

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



1. **Проект программы с определением замкнутых программных единиц и необходимых структур данных**

Программа состоит из одного блока цикла, одного блока ввода и трех блоков вывода. Структура данных – нет.

1. **Текст программы:**

**на языке С++**

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

double a\_n=0,eps=0,sum = 0.0;

int n = 1;

//ввод eps

a\_n = 1.0/((3.0\*n-2.0)\*(3.0\*(n++)+1.0));

cout << "Enter epsilon( less then 1):"; cin >> eps;

//вычисление и суммирование членов последовательности

while(abs(a\_n) >= eps)

{

sum += a\_n;

cout << "a\_" << n-1 << " = " << a\_n << endl;

a\_n = 1.0/((3.0\*n-2.0)\*(3.0\*(n++)+1.0));

}

cout <<"Sum is "<< sum;

return 0;

}

**на языке Pascal**

var

a\_n,sum,eps:real;

n: integer;

begin

sum:= 0.0;

n := 1;

a\_n := 1.0/((3.0\*n-2.0)\*(3.0\*n+1.0));

n := n + 1;

//ввод eps

writeln('Enter epsilon( less then 1):'); readln(eps);

//вычисление и суммирование членов последовательности

while abs(a\_n) >= eps do

begin

sum := sum + a\_n;

writeln('a\_',n-1,' = ', a\_n);

a\_n := 1.0/((3.0\*n-2.0)\*(3.0\*n+1.0));

n := n + 1;

end;

writeln('Sum is ', sum);

end.

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

Введем : 0.01, результат 0.3.

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

С++: 0.17 сек, pascal: 0.36 сек. Данная программа на C++ выполняется в среднем 2 раз быстрее чем на pascal

Контрольные вопросы:

1. Может ли цикл while имитировать цикл do-while? Ответ пояснить.

- Эти циклы отличаются только тем, что второй гарантированно выполняет первую итерацию, если сделать такое условие для первого цикла, чтобы в любом случае был совершен вход в цикл, то со следующих итераций эти циклы будут почти идентичны

1. Пояснить, как работает открытый цикл в С++. Для чего он может быть использован

- бесконечный цикл имеет пустое тело, следовательно он будет выполнятся до тех пор, пока не будет совершен принудительных выход каким-либо способом. Может использоваться для выполнения неизвестного кол-ва раз, пока не будет удовлетворено некое условие.

3) В чем заключаются отличия циклов с постусловием в ЯП С++ и Pascal?

- в ключевых словах, и отсутствием необходимости в rep-until использовать опер. скобки.

4) Почему в цикле repeat-until в теле цикла даже при наличии составного оператора не требуются операторные скобки

- repeat until сами действуют как опер.скобки