Федеральное государственное автономное

образовательное учреждение

высшего профессионального образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5**

(вариант 7)

Тема работы

Разработка и отладка программ циклического вычислительного процесса

Преподаватель \_\_\_\_\_\_\_\_\_\_\_

Покидышева Л. И.

подпись, дата

Студент гр. КИ-15 08 Б, Войченко В.В.

\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

Красноярск 2015

**Цель работы**

•изучить операторы цикла;

•овладеть навыками разработки циклических программ;

•получить навыки по отладке и тестированию программ;

**Задание 1**

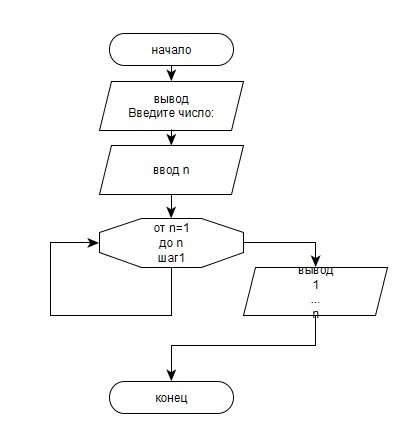
1.  Разработать графическую схему алгоритма и программу, использующую оператор цикла со счетчиком, которая для заданного натурального числа N, выведет в столбик все числа от 1 до N.

2.  Выполнить пункты 2-4 из упр.№1. Провести трассировку программы с помощью встроенного в среду программирования отладчика, анализируя значения переменных после каждого оператора присваивания.

3.  Модифицировать программу, заменив  оператор цикла со счетчиком сначала оператором цикла с предусловием, а затем с постусловием. Провести трассировку программы.  Проанализировать полученные результаты.

4.  Выполнить несколько запусков программы для заранее подготовленных тестовых примеров для проверки работоспособности всех ветвей программы. Полученные результаты проанализировать.

**Схема алгоритма программы 1**



**Исходный код программы 1**

#include <stdio.h>

#include <iostream>

#include <clocale>

#include <cstdlib>

using namespace std;

int main ()

{

int n;

setlocale (LC\_ALL, "russian");

printf ("Введите число:\n\n");

cin>>n;

cout<<""<<endl;

while (n<0)

{

cout<<"Число не может быть отрицательным!"<<endl;

cout<<"Повторите ввод! \n"<<endl;

cin>>n;

cout<<"\n"<<endl;

}

/\*for (int i=1; i<n+1; i++) //СЧЕТЧИК

cout<<i<<endl;

\*/

/\*int i=1;

do

{

cout<<i<<endl;

i=i+1;

}

while (i<n+1); //ПОСТУСЛОВИЕ

\*/

int i=1;

while (i<n+1)

{

cout<<i<<endl //ПРЕДУСЛОВИЕ

i=i+1;

}

cout<<""<<endl;

system ("pause");

return 0;

cout <<"Спасибо за пользование! \n\n";

}

**Таблица трассировки счетчика программы 1:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № строки прогаммы |  | | | | |
| 13 | n=2 |  |  |  |  |
| 25 |  | i=1 |  |  |  |
| 25 |  |  | 1)( i<n+1) да  i++ (2)  2)(i<n+1)да  i++ (3)  3) (i<n+1) нет  выход из цикла |  |  |
| 26 |  |  |  | 1)вывод i (1)  2)вывод i (2) |  |

**Таблица трассировки оператора цикла с постусловием программы 1:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № строки прогаммы |  | | | | |
| 13 | n=2 |  |  |  |  |
| 20 |  | i=1 |  |  |  |
| 23 |  |  | 1)вывод i(1)  2) вывод i(2) |  |  |
| 24 |  |  |  | 1)i=i+1(2)  2)i=i+1 (3) |  |
| 17 |  |  |  |  | 1)(i<n+1) да  2)(i<n+1) нет  выход из цикла |

**Таблица трассировки оператора цикла с предусловием программы 1:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № строки прогаммы |  | | | | |
| 13 | n=2 |  |  |  |  |
| 29 |  | i=1 |  |  |  |
| 30 |  |  | 1)(i<n+1) да  2)(i<n+1) да  3)(i<n+1) нет  выход  из цикла |  |  |
| 32 |  |  |  | 1)вывод i (1)  2) вывод i(2) |  |
| 33 |  |  |  |  | 1)i=i+1 (2)  2)i=i+1 (3) |

**Таблица тестовых наборов данных для программы 1:**

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 1 | 1 |
| 2 | 1  2 |
| 3 | 1  2  3 |
| -5 | Число не может быть отрицательным!  Повторите ввод! |
| 0 | <программа ничего не выведет> |

**Задание 2**

  Составить программу, которая бы проверяла, являются ли три натуральных числа a, b, c взаимно простыми. Числа называются взаимно простыми, если их наибольший общий делитель (НОД) равен 1. Значит, для решения этой задачи будем  дважды находить НОД чисел: НОД(a, b), а затем найти НОД(НОД(a, b), c). Нахождение НОД оформить как функцию.

**Схема алгоритма программы 2**

**Исходный код** **программы 2**

#include <iostream>

#include <cmath>

#include <clocale>

#include <cstdlib>

using namespace std;

long NOD (long a, long b, long c);

int main ()

{

long a, b, c;

if ((a<0)||(b<0)||(c<0)) //НОД определен для ВСЕХ натуральных чисел

{

a=abs(a);

a=abs(b);

a=abs(c);

}

setlocale (LC\_ALL, "russian");

cout<<"Введите число à:"<<endl;

cin>>a;

cout<<" Введите число b:"<<endl;

cin>>b;

cout<<" Введите число c:"<<endl;

cin>>c;

cout<<endl;

NOD(a, b, c);

if ((a==b==1) || (a==c==1) || (c==b==1))

cout<<"Числа взаимно простые \n\n";

else

cout<<"Числа НЕ взаимно простые \n\n";

cout<<"Спасибо за пользование! \n\n";

system ("pause");

return 0;

}

long NOD (long a, long b, long c)

{

int checkA=5;

while (a && b)

if (a>=b)

{

a%=b;

checkA=a;

checkA=0;

return b;

}

else

{

b%=a;

return a;

}

if (checkA==0)

{

while (b && c)

if (b>=c)

{

b%=c;

return c;

}

while (b && c)

if (b<=c)

{

c%=b;

return b;

}

}

else

{

while (b && c)

if (a>=c)

{

a%=c; return c;

}

}

if (checkA==5)

{

while (b&&c)

if (a<=c)

{

c%=a;

return a;

}

}

**Тестовые наборы данных для программы 2:**

|  |  |
| --- | --- |
| Входные данные | Результат |
| 7 7 7 | Числа взаимно простые |
| 5 10 25 | Числа НЕ взаимно простые |
| 0 0 0 | Числа взаимно простые |
| 100 10 1000 | Числа НЕ взаимно простые |