


Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
Томский государственный университет систем управления
и радиоэлектроники (ТУСУР)
Кафедра экономической математики, информатики и статистики (ЭМИС)

ОПЕРАТОРЫ ЦИКЛА В ЯЗЫКЕ СИ
Отчёт по лабораторной работе по дисциплине
«Введение в программирование»

Студент гр. 543-1
 /А.Е. Мухамеджан

“ ____ ” ____ 2023

Руководитель доцент
кафедры ЭМИС

____ Е.А. Шельмина

“ ____ ” ____ 2023г.

Томск 2023

Лабораторная работа №4

Операторы цикла в языке Си

Цель работы: изучить операторы цикла языка программирования Си.

Теоретические сведения

Циклические вычислительные процессы

Очень часто в программировании бывают случаи, когда требуется многократное повторение одних и тех же действий. Для того, чтобы не было необходимости в многократном повторении одинаковых фрагментов кода программы, предусмотрены специальные операторы циклического повторения выделенного фрагмента программы.

Существует несколько различных операторов, предназначенных для организации циклов:

- *операторы цикла с предусловием,*
- *оператор цикла с постусловием.*

Циклы с предусловием, когда решение об очередном выполнении принимается перед началом его выполнения.

Для организации цикла необходимо:

- *Выбрать переменную, которая, меняясь, будет менять значение условного выражения, и проинициализировать ее. Такая переменная называется управляющей переменной цикла.*
- *Выбрать условное выражение, обеспечивающее повторение тела цикла или прекращение его выполнения.*
- *Определить, какие действия должны выполняться при каждом повторении, т.е. запрограммировать тело цикла.*
- *Внутри цикла произвести коррекцию (изменение) управляющей переменной цикла.*

При этом инициализация управляющей переменной должна быть выполнена до цикла (вне его).

Оператор цикла *while* предназначен для организации цикла с предусловием. Число повторов определяется тем, выполнено или нет условие, которое стоит в начале (заголовке) цикла.

Блок-схема такого цикла представлена на рисунке 1.

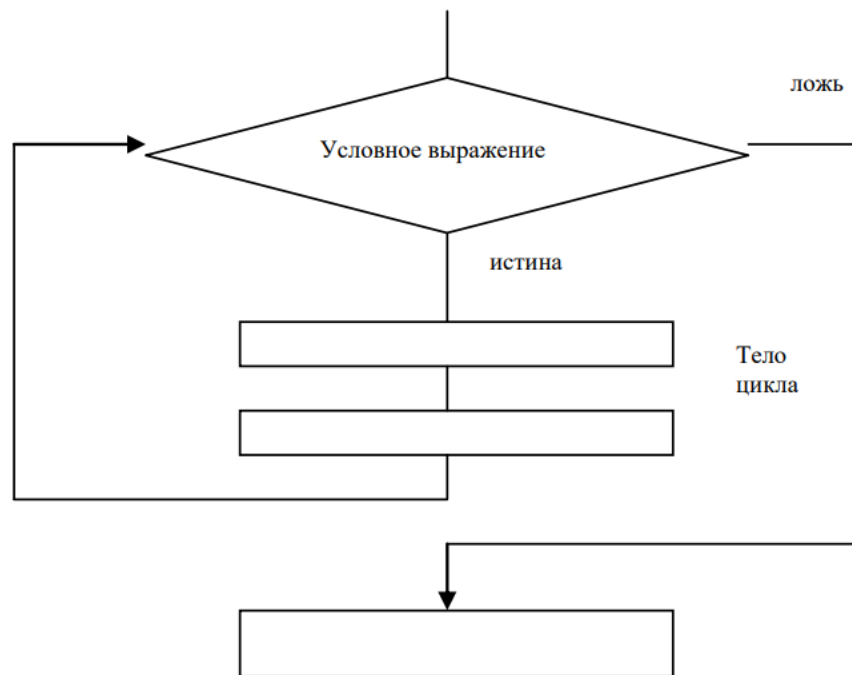


Рисунок 1 – Скриншот блок-схемы цикла с предусловием.

Формат:

```
while ( условное выражение )  
    тело цикла;
```

Правило выполнения: Цикл повторяется до тех пор, пока выполняется заданное условие, т.е. условие истинно (не равно 0).

Второй оператор цикла с предусловием оператор *for* предназначен для организации цикла с заданным числом повторов. Число повторов задается в параметрах цикла. Циклы данного типа имеют так называемый счетчик. Требуется задать его начальное и конечное значения и шаг изменения. В отличие от предыдущих операторов цикла, где инициализация, проверка условия и коррекция выполняется в разных местах, в операторе *for* они собраны в одном месте.

Формат оператора *for*:

for (инициализация; условие; коррекция)
 тело цикла;

Правило выполнения:

- Инициализация выполняется только один раз (при первом вхождении в цикл).
- Проверка условного выражения. Если условие истинно, то к п.3, иначе – выход из цикла.
- Выполняются операторы тела цикла.
- Выполняется коррекция, после этого переходим к пункту 2, и так до тех пор, пока условие не станет ложным. Тогда управление передается вниз, на следующий за оператором *for* оператор.

Циклы бывают с постусловием, когда решение об очередном выполнении принимается после каждой итерации (выполнения) тела цикла. В этом случае операторы тела цикла выполняются хотя бы один раз.

Блок-схема такого цикла представлена на рисунке 2.

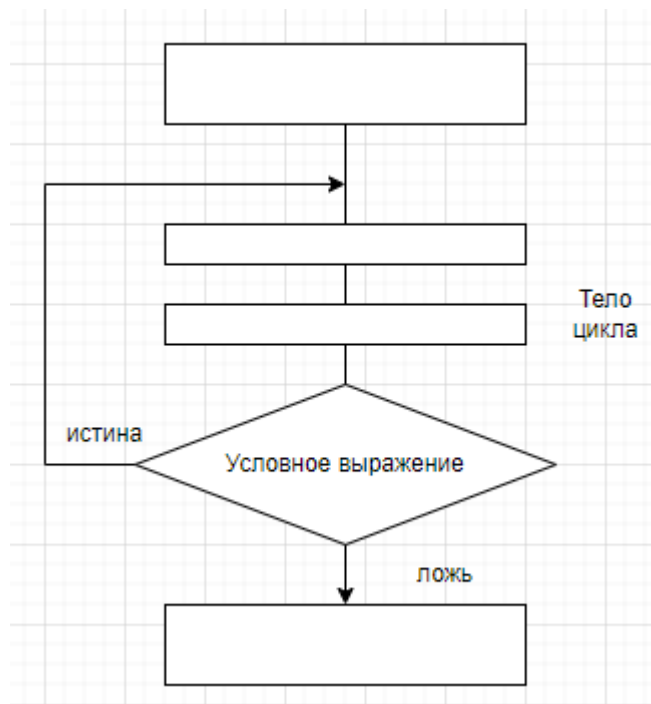


Рисунок 2 – Скриншот блок-схемы цикла с постусловием

Оператор цикла *do ... while* предназначен для организации цикла с постусловием. Число повторов определяется тем, выполнено или нет условие, которое стоит в конце цикла.

Отличие от цикла типа *while* состоит в том, что у цикла *while* сначала проверяется условие, а затем выполняется очередной повтор тела цикла (т.е. возможен случай, когда тело цикла не выполняется ни разу), а у цикла типа *do ... while* сначала выполняется тело цикла, а потом проверяется условие и в зависимости от него начинается повтор или нет (т.е. тело цикла всегда выполняется хотя бы 1 раз).

Формат:

```
do
{
    тело цикла;
}
while ( условное выражение );
```

Правило выполнения: Цикл повторяется до тех пор, пока выполняется заданное условие. Правила написания условий, такие же, как и в операторе *if ... else*.

Типичный пример циклического процесса — вычисление конечных сумм. При определении сумм многократно вычисляется выражение, стоящее под знаком суммы и складывается с предыдущей частичной суммой. Вычисления производятся до тех пор, пока не будут сложены выражения под знаком суммы для всех значений изменяющейся переменной.

Например, необходимо составить программу, вычисляющую значение суммы:

$$S = \sum_{i=1}^{10} \frac{i}{i+1}.$$

Прежде чем вычислять выражение под знаком суммы необходимо определить начальные значения суммы $S = 0$ и начальное значение параметра цикла $i = 1$.

Затем вычисляется выражение под знаком суммы для $i = 1$ и складывается с предыдущим значением суммы S . При этом в переменной S содержится i -я частичная сумма. После этого значение i увеличивается на единицу и проверяется, не стало ли оно больше 10. Если значение i меньше или равно 10, то вычисляется следующая частичная сумма. В противном случае вычисление суммы будет закончено. Конечное значение суммы S выводится на экран.

Так как количество повторений заранее известно, то целесообразно использовать оператор цикла `for`, включающий в себя блоки инициализации, проверки условия продолжения цикла, изменения переменной цикла после выполнения операторов тела цикла. На рисунке 3 показана простейшая блок-схема вычисления суммы при помощи цикла `for`. В блоке цикла указывается имя параметра цикла, его начальное и конечное значения, шаг изменения (если шаг равен 1 его можно не указывать).

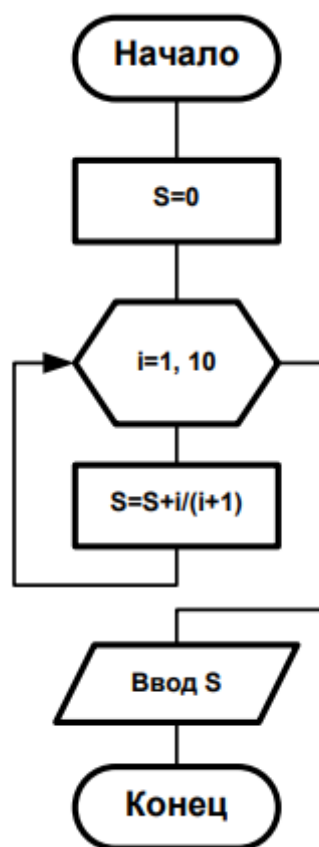


Рисунок 3 – Скриншот блок-схемы вычисления суммы при помощи цикла `for`

Ход работы

Задание 1. Написать программу на языке Си для вычисления конечной суммы, результаты расчета представить в виде таблицы.

Условие к заданию 1 представлено на рисунке 4.

12	$\sum_{m=1}^{16} \sqrt{\frac{m^{2,5} + 1}{2m}}$
----	---

Рисунок 4 – Скриншот условия к заданию №1

Программа к заданию №1 представлена на рисунке 5.

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <math.h>
5  #include <locale.h>
6
7  int main()
8  {
9      setlocale(LC_CTYPE, "Rus");
10     float sum = 0;
11     for (int m = 1; m <= 16; m++) {
12         //printf("%i\n", m);
13         sum += (sqrt((pow(m, 2.5) + 1) / 2 * m));
14         printf("%i\t%5.5f\t%5.5f\n", m, (sqrt((pow(m, 2.5) + 1) / 2 * m)), sum)
15     }
16 }
```

Рисунок 5 – Скриншот программы к заданию №1

Вывод программы представлен на рисунке 6.

```
Консоль отладки Microsoft Visual Studio

1      1.00000 1.00000
2      2.58009 3.58009
3      4.98825 8.56834
4      8.12404 16.69238
5      11.92704 28.61942
6      16.35680 44.97622
7      21.38332 66.35954
8      26.98291 93.34245
9      33.13608 126.47853
10     39.82636 166.30489
11     47.03949 213.34438
12     54.76298 268.10736
13     62.98570 331.09305
14     71.69766 402.79071
15     80.88980 483.68051
16     90.55385 574.23438

C:\Users\mr101\source\repos\lab3\x64\D
Чтобы автоматически закрывать консоль
томатически закрыть консоль при остано
Нажмите любую клавишу, чтобы закрыть э
```

Рисунок 6 – Скриншот вывода программы к заданию №1

Задание 2. Написать программу на языке Си для вычисления бесконечной суммы с заданной точностью.

Условие к заданию №2 представлено на рисунке 7.

12	$\sum_{k=1}^{\infty} \frac{e^{kx} - x}{k!}$	1,21	$3 \cdot 10^{-5}$
----	---	------	-------------------

Рисунок 7 – Скриншот условия к заданию №2

Программа к заданию №2 представлена на рисунке 8.

```

1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <math.h>
5  #include <locale.h>
6  int main()
7  {
8      setlocale(LC_CTYPE, "Rus");
9      const double x = 1.21, e = 0.00003;
10     int k = 1; double sum = 0;
11     do {
12         double fact = k;
13         for (int i = 1; i < k; i++) {
14             fact *= i;
15         }
16         sum += (exp( k * x) - x) / fact;
17         k++;
18         printf("%.5f\n", sum);
19     } while (sum < e);
20 }

```

Рисунок 8 – Скриншот программы к заданию №2

Вывод программы представлен на рисунке 9.

```

2.14348
C:\Users\mr101\source\repos\ConsoleApplication4\x64\Debug\ConsoleApplication4.exe (процесс 16992) завершил работу с кодо
м 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Ав
томатически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно:

```

Рисунок 9 – Скриншот вывода программы к заданию 2

Задание 3. Вычислить значения функции $f(x)$ на отрезке $[a;b]$ с шагом h .

Условие к заданию №3 представлено на рисунке 10.

12	$y=2.09x^2+1.22$	$x \in [4; 9.3]$	$h=0.3$
----	------------------	------------------	---------

Рисунок 10 – Скриншот условия к заданию №3

Программа к заданию №3 представлена на рисунке 11.

```

1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <math.h>
5  #include <locale.h>
6  int main()
7  {
8      setlocale(LC_CTYPE, "Rus");
9      float x = 4;
10     while(x <= 9.3){
11         float y = 2.09 * pow(x, 2) + 1.22;
12         printf("%f - %f\n-----\n", x, y);
13         x += 0.3;
14     }
15 }
```

Рисунок 11 – Скриншот программы к заданию №3

Вывод программы представлен на рисунке 12.

```

4.000000 - 34.660000
-----
4.300000 - 39.864105
-----
4.600000 - 45.444408
-----
4.900001 - 51.400913
-----
5.200001 - 57.733616
-----
5.500001 - 64.442520
-----
5.800001 - 71.527626
-----
6.100001 - 78.988937
-----
6.400002 - 86.826439
-----
6.700002 - 95.040146
-----
7.000002 - 103.630058
-----
7.300002 - 112.596161
-----
7.600002 - 121.938469
-----
7.900002 - 131.656982
-----
8.200003 - 141.751694
-----
8.500003 - 152.222595
-----
8.800003 - 163.069717
-----
9.100003 - 174.293030
-----
```

Рисунок 12 – Скриншот вывода программы к заданию №3

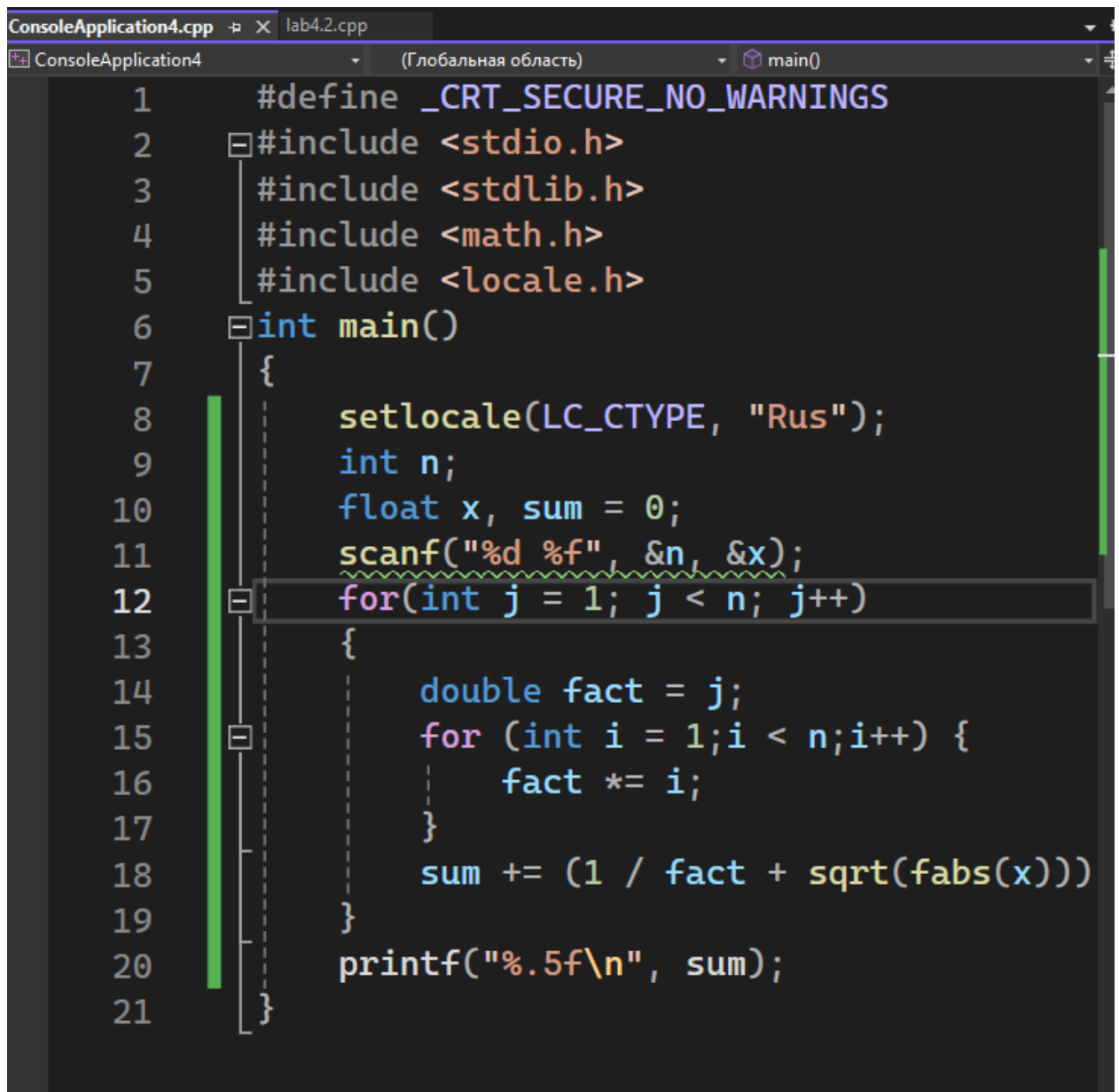
Задание 4. Дано натуральное n и действительное x . Вычислить значение суммы.

Условие к заданию №4 представлено на рисунке 13.

2	$\left(\frac{1}{1!} + \sqrt{ x }\right) + \left(\frac{1}{2!} + \sqrt{ x }\right) + \dots + \left(\frac{1}{n!} + \sqrt{ x }\right)$
---	--

Рисунок 13 – Скриншот условия к заданию №4

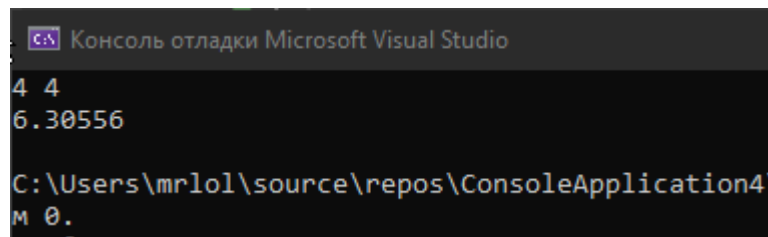
Программа к заданию №4 представлена на рисунке 14.



```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <math.h>
5  #include <locale.h>
6  int main()
7  {
8      setlocale(LC_CTYPE, "Rus");
9      int n;
10     float x, sum = 0;
11     scanf("%d %f", &n, &x);
12     for(int j = 1; j < n; j++)
13     {
14         double fact = j;
15         for (int i = 1; i < n; i++) {
16             fact *= i;
17         }
18         sum += (1 / fact + sqrt(fabs(x)))
19     }
20     printf("%.5f\n", sum);
21 }
```

Рисунок 14 – Скриншот программы к заданию №4

Вывод программы представлен на рисунке 15.



```
Консоль отладки Microsoft Visual Studio
4 4
6.30556
C:\Users\mr101\source\repos\ConsoleApplication4
м 0.
```

Рисунок 15 – Скриншот вывода программы к заданию №4

Вывод: в ходе лабораторной работы №4 изучены операторы цикла языка программирования Си, выполнены предоставленные задания.