


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 ВЕТВЛЕНИЯ И ЛОГИЧЕСКИЕ ВЫРАЖЕНИЯ В ЯЗЫКЕ
C/C++

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 №3

Операторы ветвления и логические выражения в языке C/C++

Цель работы: изучить операторы условия, операции отношения, логические операции и выражения языка программирования Си.

Теоретические сведения

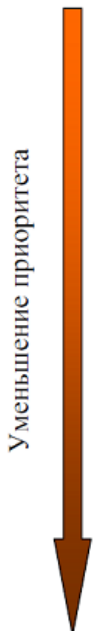
Операции сравнения применяются при работе с двумя операндами и возвращают true (1), если результат сравнения – истина, и false (0), если результат сравнения – ложь. В языке Си определены следующие операции сравнения: < (меньше), <= (меньше или равно), > (больше), >= (больше или равно), != (не равно), == (равно).

Логические операции работают с операндами скалярных типов и возвращают результат булева типа. Существует три логические операции:

- ! – отрицание, или логическое НЕ
- && - логическое И
- || - логическое ИЛИ.

Приоритет операций в C/C++ представлен на рисунке 1.

Приоритет операций в C/C++	
Тип операции	Операторы
Разрешения области действия	::
Другие	[], (), «.» (точка),
Унитарные	&, +, -, !, префикс ++ и --
Унитарные	*, ->
Арифметические	*, /, %
Арифметические	+, -
Унитарные	постфикс ++ и --
Поразрядный сдвиг	<<, >>
Сравнение	>, <, >=, <=
Сравнение	==, !=
Поразрядные логические	&
Поразрядные логические	^
Поразрядные логические	
Логические	&&
Логические	
Условная	?:
Присваивания	=, *=, /=, %=, +=, -=, <<=, >>=, &=, ^=, =
Последовательность	«,» (запятая)



Уменьшение приоритета

Рисунок 1 – Скриншот таблицы приоритета операций в C/C++

Оператор if else

Когда требуется не просто выполнять операции последовательно (линейный процесс), а изменять ход вычислительного процесса в зависимости от некоторых условий, используется **ветвление по условию**

Формат оператора if ... else :

БЛОК 1;

if (условие)

{ БЛОК 2; }

else

{ БЛОК 3; }

БЛОК 4;

Блок-схема оператора представлена на рисунке 2.

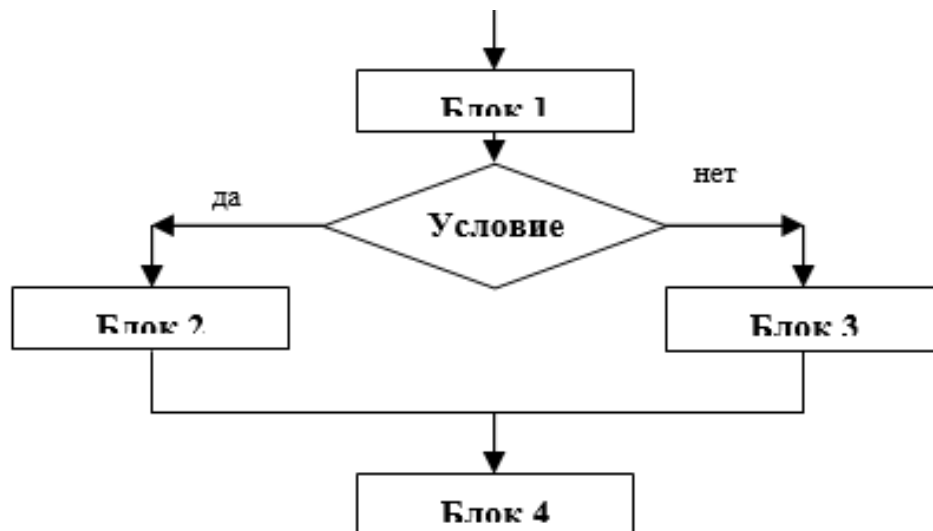


Рисунок 2 – Скриншот блок-схемы оператора условия

Правило выполнения: сначала выполняется БЛОК 1, затем, если указанное условие истинно, то выполняется БЛОК 2, если ложно, то выполняется БЛОК 3. Затем выполняется БЛОК 4. Т.е. в зависимости от условия выполняется либо БЛОК 2, либо БЛОК 3.

Итак, все зависит от выполнения или невыполнения условия. Обычно условия могут состоять из переменных, выражений и функций. Единственное требование состоит в том, что сравнивать можно сравнимое.

Любое выражение, в том числе логическое, всегда имеет значение. Если выражение ложно, то оно равно 0. Истинным является выражение, не равное 0 (положительные, отрицательные, целые, дробные).

Для записи условий применяются специальные знаки представленные на рисунке 3.

==	Равно
<	Меньше
<=	Меньше или равно
>	Больше
>=	Больше или равно
!=	Не равно
&&	И
	ИЛИ

Рисунок 3 – Скриншот таблицы специальных знаков для записи условий

Оператор - переключатель *switch()*

Для тех случаев, когда требуется выбрать действия не из двух вариантов, а из нескольких, предназначен оператор переключатель *switch*.

Конечно, можно было бы использовать несколько операторов *if ... else*, но оператор *switch* оказывается намного удобнее. Он позволяет обработать сразу несколько сравнений по условию «выражение равно значению».

Формат:

switch (параметр-выражение)

```
{
    case 1_значение: { блок 1          }
    case 2_значение: { блок 2 break ; }
    case 3_значение: { блок 3 break ; }
    case 4_значение:
    case 5_значение:
    case 6_значение: { блок 4 break ; }
    case 7_значение: { блок 5          }
```

```

case 8_значение: { блок 6 break ; }
и т.д.
default          : { блок 7 }
}

```

Значения 1-8 – константные выражения целочисленного типа или char, которые сравниваются с параметром-выражением. В качестве параметра может использоваться только переменная целого или символьного типа или целочисленное выражение, содержащее переменные данного типа.

Правило выполнения: Выполняется блок, помеченный значением, которое оказывается равным параметру-выражению. А следом выполняются все последующие блоки, пока не встретится специальный оператор break. Его смысл состоит в том, что он прекращает выполнение главного оператора, внутри которого он находится.

Правило выполнения. (По примеру)

Если параметр равен 1 значению, то выполняются блок 1 и блок 2.

Если параметр равен 2 значению, то выполняется блок 2.

Если параметр равен 3 значению, то выполняется блок 3.

Если параметр равен 4 значению,

или 5 значению,

или 6 значению, то выполняется блок 4.

Если параметр равен 7 значению, то выполняется блок 5 и блок 6.

Если параметр равен 8 значению, то выполняется блок 6.

и т.д.

Если параметр не равен ни одному из заданных значений, то выполняется раздел default т.е. блок 7.

Ход работы

Задание 1. Разработать блок-схему и написать программу на языке C/C++ для решения следующей задачи: вычислить значение функции. Значение x вводить с клавиатуры, используя оператор *scanf()*. Если x не

принадлежит вышеприведенным числовым интервалам, то вывести на экран соответствующее сообщение. В противном случае вывести значение y , используя функцию *printf* ().

Условие к заданию №1 представлено на рисунке 4.

$y = \begin{cases} 1.5x^{1.2} - 4.22 & x \leq 0 \\ -x^{3.22} + x^{2.1} - x^{4.1} - 2 & 5.1 < x < 8.2 \\ \frac{\sin(x+2)}{10-x} & 0 \leq x \leq 5 \end{cases}$

Рисунок 4 – Условие к заданию №1

Блок-схема к заданию №1 представлена на рисунке 5.

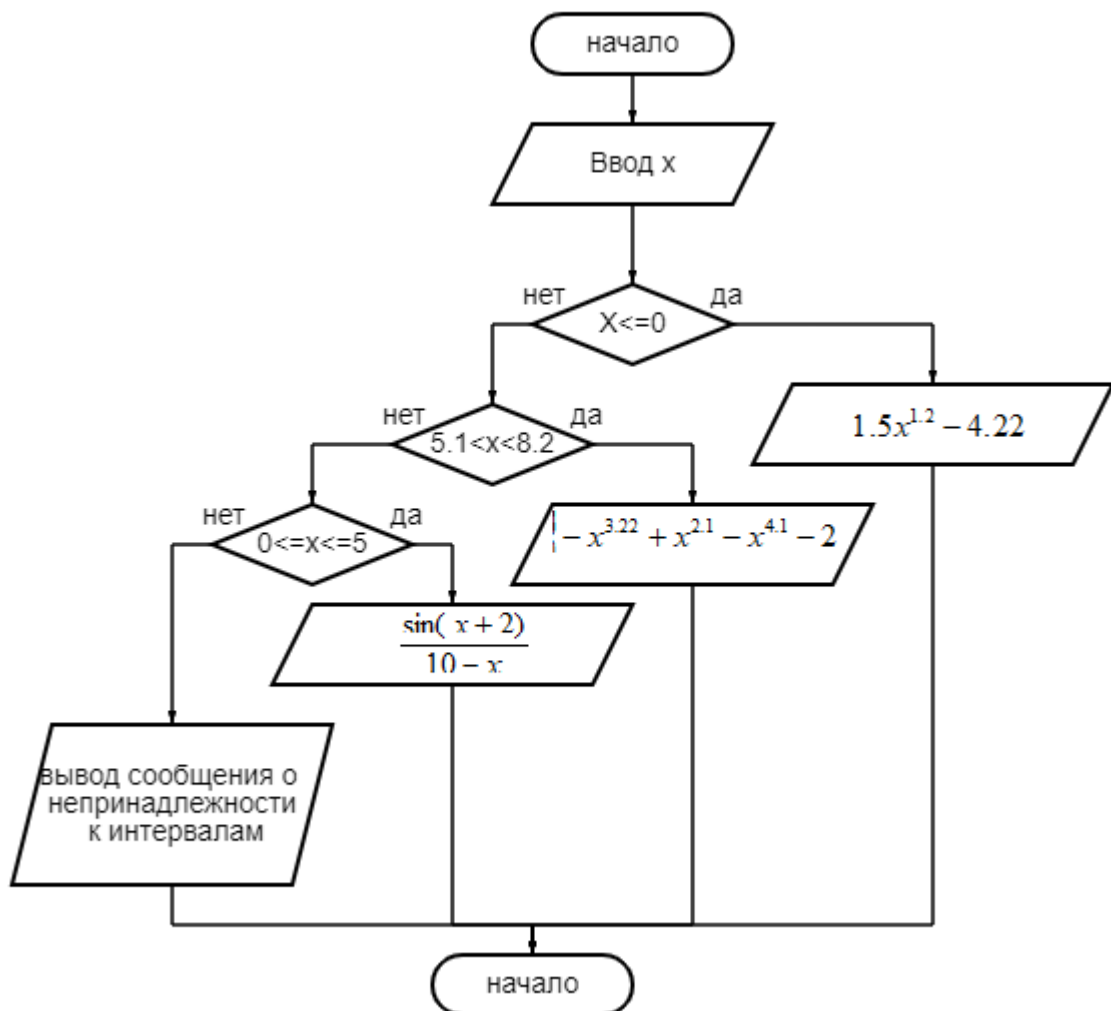


Рисунок 5 - Блок-схема к заданию №1

Программа к заданию №1 представлена на рисунке 6.

```

1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <math.h>
5
6  int main()
7  {
8      float x;
9      printf("x:");
10     scanf("%f", &x);
11     if (x <= 0) {
12         printf("%f\n", (1.5 * pow(x, 1.2) - 4.22));
13     }
14     else if (x > 5.1 && x < 8.2) {
15         printf("%f\n", (-pow(x, 3.22) + pow(x, 2.1) - pow(x, 4.1) - 2));
16     }
17     else if (x >= 0 && x <= 5) {
18         printf("%f\n", (sin(x + 2) / (10 - x)));
19     }
20     else {
21         printf("x = (-inf;8.2)");
22     }
23 }

```

Рисунок 6 – Программа к заданию №1

Вывод программы к заданию №1 представлен на рисунке 7.

```

k:3
-0.136989
C:\Users\mr1ol\source\repos\ConsoleApplication4\x64\Debug\ConsoleApplication4.exe (процесс 13556) завершил работу с кодо
м 0.

```

Рисунок 7 – Вывод программы к заданию №1

Задание 2. Написать программу вычисления значения функции y в зависимости от введенного с клавиатуры значения переменной i , используя оператор switch. Необходимые данные вводить по мере необходимости.

Условие к заданию №2 представлено на рисунке 8.

12	$y = 2^{a^b} + \frac{7.02}{(a + \frac{a}{b})} \sin(2a) \quad \text{при } a = 1.103$ <p>..... $b = 0.257$</p> $y = 2^{a^b} + \frac{7.02}{(a + \frac{a}{b})} \sin(2a) \quad \text{при } a = 1.103$ <p>..... $b = 0.257$</p> <p>в других случаях $y = \begin{cases} \frac{1.5x}{x+1} & x \geq 10 \\ -3.5x & 2 < x < 10 \\ -\sqrt{5x} & x \leq 2 \end{cases}$</p>
----	--

Рисунок 8 – Условие к заданию №2

Программа к заданию №2 представлена на рисунке 9.

```

6 int main()
7 {
8     int i;
9     float a = 1.103, b = 0.257;
10    printf("i:");
11    scanf("%i", &i);
12    switch (i) {
13        case 1:
14        case 10:
15        printf("y = %f", (pow(2, pow(a, b)) + 7.02 / (a + a / b) * sin(2 * a)));
16        break;
17
18        default:
19            if (i >= 10) {
20                printf("y = %f", (1.5 * i / (i + 1)));
21            }
22            else if (i > 2 && i < 10) {
23                printf("y = %f", (3.5 * i));
24            }
25            else if (i <= 2) {
26                printf("y = %f", (-sqrt(5 * i)));
27            }
28    }
29 }

```

Рисунок 9 – Программа к заданию №2

Вывод программы к заданию №2 представлен на рисунке 10.

```

Консоль отладки Microsoft Visual Studio
y = -3.162278
C:\Users\mr101\source\repos\ConsoleApplication4\x64\Debug\ConsoleApplication
4.exe (процесс 21560) завершил работу с кодом 0.

```

Рисунок 10 – Вывод программы к заданию №2

Задание 3. Разработать блок-схему и написать программу на C/C++ для решения задачи согласно варианту.

Условие к заданию №3 представлено на рисунке 11.

Вариант 12. Дано a, b . Найти наибольшее $\max(a, b)$ и наименьшее $\min(a, b)$ из чисел a и b . Найти d :

$$d = \frac{(\max(a, b))^5 - 3.2 \cdot \min(a, b)}{1 + \min(a, b)}$$

Рисунок 11 - Условие к заданию №3

Блок-схема к заданию №3 представлена на рисунке 12

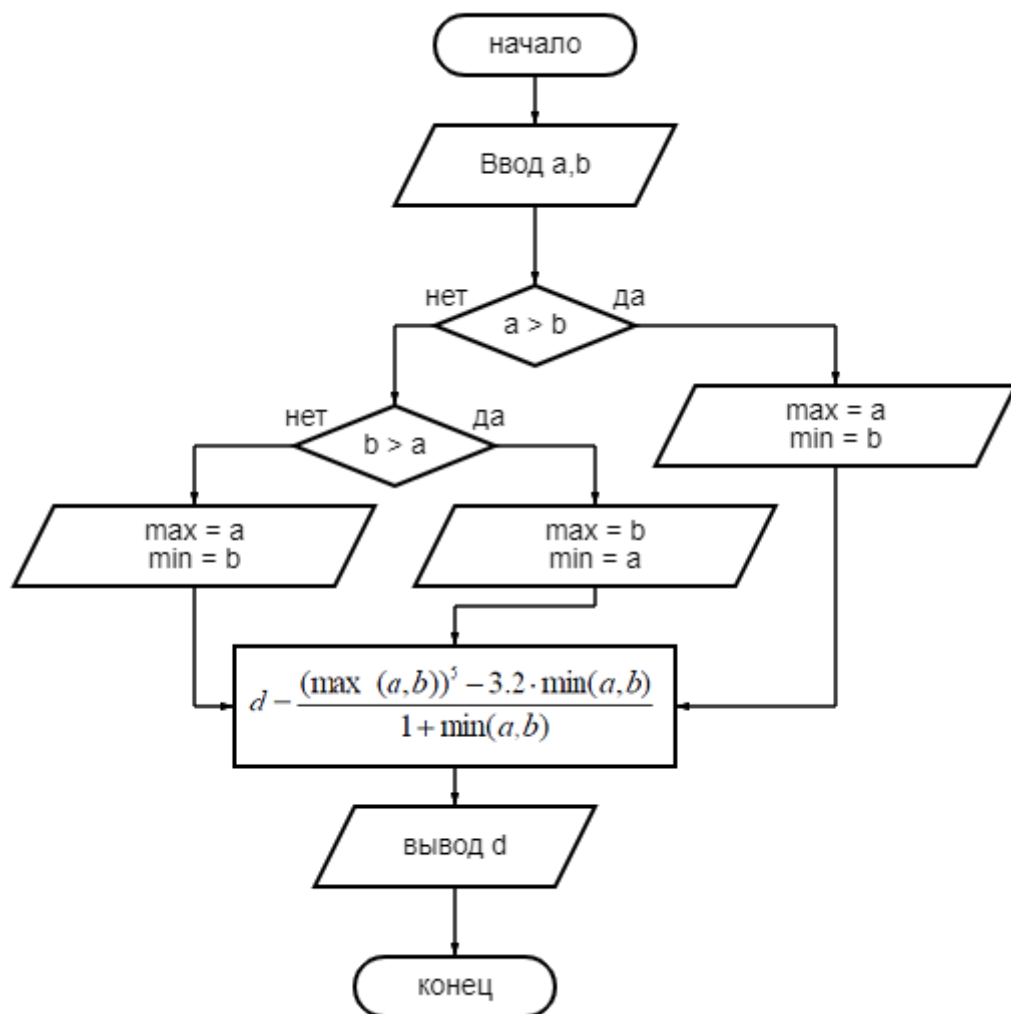


Рисунок 12 - Блок-схема к заданию №3

Программа к заданию №3 представлена на рисунке 13.

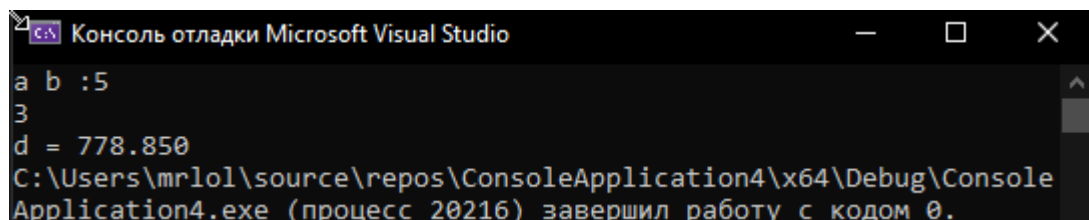
```

4  | #include <math.h>
5  |
6  | int main()
7  | {
8  |     float a, b, max, min;
9  |     printf("a b :");
10 |     scanf("%f%f", &a, &b);
11 |     if (a > b) {
12 |         max = a;
13 |         min = b;
14 |     }
15 |     else if (b > a) {
16 |         max = b;
17 |         min = a;
18 |     }
19 |     else {
20 |         max = a;
21 |         min = b;
22 |     }
23 |     printf("d = %5.3f", ((pow(max, 5) - 3.2 * min) / (1 + min)));
24 | }

```

Рисунок 13 – Программа к заданию №3

Вывод программы к заданию №3 представлен на рисунке 14.



```

Консоль отладки Microsoft Visual Studio
a b :5
3
d = 778.850
C:\Users\mr1ol\source\repos\ConsoleApplication4\x64\Debug\Console
Application4.exe (процесс 20216) завершил работу с кодом 0.

```

Рисунок 14 – Вывод программы к заданию №3

Задание 4. Написать программу на C/C++ для решения задачи согласно варианту. Использовать оператор **switch**.

Условие к заданию №4 представлено на рисунке 15.

Вариант 12. Дан номер месяца - целое число в диапазоне 1-12 (1 - январь, 2 - февраль и т. д.). Определить количество дней в этом месяце для невисокосного года. Использовать оператор выбора **switch**.

Рисунок 15 - Условие к заданию №4

Программа к заданию №4 представлена на рисунке 16.

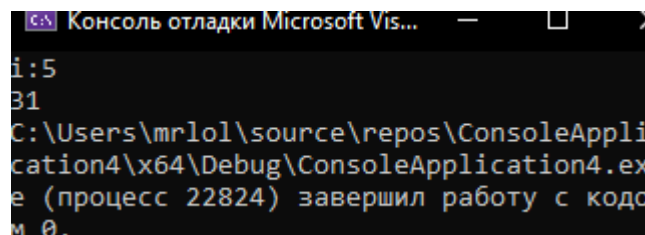
```

4      [ #include <math.h>
5
6      [ int main()
7      {
8          int i;
9          printf("i:");
10         scanf("%i", &i);
11         switch (i) {
12             case 1:
13             case 3:
14             case 5:
15             case 7:
16             case 8:
17             case 10:
18             case 12:
19                 printf("31");
20                 break;
21             case 4:
22             case 6:
23             case 9:
24             case 11:
25                 printf("30");
26                 break;
27             case 2:
28                 printf("28");
29         }
30     }

```

Рисунок 16 – Программа к заданию №4

Вывод программы к заданию №4 представлен на рисунке 17.



```

i:5
31
C:\Users\mr1ol\source\repos\ConsoleAppli
cation4\x64\Debug\ConsoleApplication4.ex
e (процесс 22824) завершил работу с кодо
м 0.

```

Рисунок 7 – Вывод программы к заданию №1

Задание 5. Для решения приведенных ниже задач написать программу на языке C/C++ с использованием логического типа *bool*.

Условие к заданию №5 представлено на рисунке 18.

№1

Вычислить значение логического выражения при всех возможных значениях логических величин X, Y и Z:

а) не (X или Y) и (не X или не Z);

б) не (не X и Y) или (X и не Z);

в) X или не Y и не (X или не Z).

Рисунок 18 - Условие к заданию №5

Программа к заданию №5 представлена на рисунке 19.

```

7  int main()
8  {
9      setlocale(LC_CTYPE, "Rus");
10     bool X = true, Y = true, Z = true;
11     char a;
12     printf("a: ");
13     scanf("%c", &a);
14     for (int i = 0; i <= 1; i++) {
15         X != X;
16         for (int j = 0; j <= 1; j++) {
17             Y != Y;
18             for (int k = 0; k <= 1; k++) {
19                 Z != Z;
20                 switch (a)
21                 {
22                     case 'a':
23                         printf("%d\n", (!(X || Y) && (!X || !Z)));
24                         break;
25
26                     case 'b':
27                         printf("%d\n", (!(X && Y) || (X && !Z)));
28                         break;
29                     case 'c':
30                         printf("%d\n", (X || !Y && !(X || !Z)));
31                         break;
32                 }
33             }
34         }
35     }
36 }

```

Рисунок 19 – Программа к заданию №5

Вывод программы к заданию №5 представлен на рисунке 20.

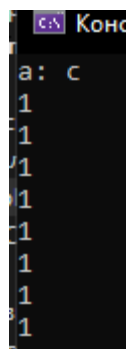


Рисунок 20 - Вывод программы к заданию №5

Вывод: в ходе выполненной лабораторной работы были изучены операторы условия, операции отношения, логические операции и выражения языка программирования Си.