

Лабораторная работа №4

Тема: «УСЛОВНЫЕ ОПЕРАТОРЫ ЯЗЫКА C»

Цель работы: изучить особенности использования условных операторов if и switch.

Теоретическая часть

Условный оператор if

Для того чтобы иметь возможность реализовать логику в программе используются условные операторы. Например, требуется определить, содержит ли некоторая переменная `arg` положительное или отрицательное число и вывести соответствующее сообщение на экран. Для этого можно воспользоваться оператором `if` (если), который и выполняет подобные проверки.

В самом простом случае синтаксис данного оператора `if` следующий:

```
if (выражение)
    <оператор>
```

Если значение параметра «выражение» равно «истинно», выполняется оператор, иначе он пропускается программой. Следует отметить, что «выражение» является условным выражением, в котором выполняется проверка некоторого условия. В табл. 2 представлены варианты простых логических выражений оператора `if`.

Таблица 2. Простые логические выражения

<code>if(a < b)</code>	Истинно, если переменная <code>a</code> меньше переменной <code>b</code> и ложно в противном случае.
<code>if(a > b)</code>	Истинно, если переменная <code>a</code> больше переменной <code>b</code> и ложно в противном случае.
<code>if(a == b)</code>	Истинно, если переменная <code>a</code> равна переменной <code>b</code> и ложно в противном случае.
<code>if(a <= b)</code>	Истинно, если переменная <code>a</code> меньше либо равна переменной <code>b</code> и ложно в противном случае.
<code>if(a >= b)</code>	Истинно, если переменная <code>a</code> больше либо равна

	переменной b и ложно в противном случае.
if(a != b)	Истинно, если переменная a не равна переменной b и ложно в противном случае.
if(a)	Истинно, если переменная a не равна нулю, и ложно в противном случае.

Приведем пример использования оператора ветвления if. Следующая программа позволяет определять знак введенной переменной.

Листинг 3. Программа определения знака введенного числа.

```
#include <stdio.h>
int main()
{
    float x;
    printf("Введите число: ");
    scanf("%f",&x);
    if(x < 0)
        printf("Введенное число %f является отрицательным.\n", x);
    if(x >= 0)
        printf("Введенное число %f является неотрицательным.\n", x);
    return 0;
}
```

Анализ приведенного текста программы показывает, что два условных оператора можно заменить одним, используя конструкцию

```
if (выражение)
    <оператор1>
else
    <оператор2>
```

которая интерпретируется таким образом. Если «выражение» истинно, то выполняется «оператор1», иначе выполняется «оператор2».

В случаях, когда при выполнении какого-либо условия необходимо записать более одного оператора, необходимо использовать фигурные скобки, т.е. использовать конструкцию вида

```

if (выражение)
{
    <список операторов>
}
else {

    <список операторов>
}

```

Следует отметить, что после ключевого слова `else` формально можно поставить еще один оператор условия `if`, в результате получим еще более гибкую конструкцию условных переходов:

```

if (выражение1) <оператор1>
else if (выражение2) <опреатор2>
else <оператор3>

```

До сих пор рассматривались простые условия типа $x < 0$. Вместе с тем оператор `if` позволяет реализовывать более сложные условные переходы. В языке C имеются три логические операции:

`&&` - логическое И `||` - логическое ИЛИ `!` – логическое НЕТ

На основе этих трех логических операций можно сформировать более сложные условия. Например, если имеются три переменные `exp1`, `exp2` и `exp3`, то они могут составлять логические конструкции, представленные в табл. 3.

Таблица 3. Пример составных логических выражений

<code>if(exp1 > exp2 && exp2 < exp3)</code>	Истинно, если значение переменной <code>exp1</code> больше значения переменной <code>exp2</code> и значение переменной <code>exp2</code> меньше значения переменной <code>exp3</code> .
<code>if(exp1 <= exp2 exp1 >= exp3)</code>	Истинно, если значение переменной <code>exp1</code> меньше либо равно значения переменной <code>exp2</code> или значение переменной <code>exp2</code> больше либо равно значения переменной <code>exp3</code> .
<code>if(exp1 && exp2 && !exp3)</code>	Истинно, если истинное значение <code>exp1</code> и истинно

	значение <code>exp2</code> и ложно значение <code>exp3</code> .
<code>if(!exp1 !exp2 && exp3)</code>	Истинно, если ложно значение <code>exp1</code> или ложно значение <code>exp2</code> и истинно значение <code>exp3</code> .

Подобно операциям умножения и сложения в математике, логические операции И ИЛИ НЕТ, также имеют свои приоритеты. Самый высокий приоритет имеет операция НЕТ, т.е. такая операция выполняется в первую очередь. Более низкий приоритет у операции И, и наконец самый малый приоритет у операции ИЛИ.

Условная тернарная операция

Условная тернарная операция используется с тремя операндами. В изображении условной операции применяются два символа '?' и ':' и три выражения – операнда:

выражение_1 ? выражение_2 : выражение_3;

Первым вычисляется значение выражения_1. Если оно истинно, т. е. не равно 0, то вычисляется значение выражения_2, которое становится результатом. Если при вычислении выражения_1 получится 0, то в качестве результата берётся значение выражения_3, например:

`x = x < 0 ? -x : x;`

Выражение возвращает абсолютную величину переменной `x`.

Пример программы определения наибольшего из значений функции $y_1 = x^2 + 1$, $y_2 = 7 - x^2$ для любого x .

```
#include <stdlib.h>
#include <stdio.h>
int main()
{
    float x, y1, y2, max;
    printf("Input x ");
    scanf("%f", &x);
    y1 = x * x + 1;
    y2 = 7 - x * x;
    max = y1 > y2 ? y1 : y2;
    printf("x=%8.2f y1=%8.2f y2=%8.2f max=%8.2f\n", x, y1,
y2, max);
    system("PAUSE");
    return 0;
}
```

Оператор выбора альтернатив switch

Условная операция `if` облегчает написание программ, в которых необходимо производить выбор между небольшим числом возможных вариантов. Однако иногда в программе необходимо осуществить выбор одного варианта из множества возможных. Формально для этого можно воспользоваться конструкцией `if else if ... else`. Однако во многих случаях оказывается более удобным применять оператор `switch` языка C. Синтаксис данного оператора следующий:

```
switch(переменная)
{
    case константа1:
        <операторы>
    case константа2:
        <операторы>
    ...
    default:
        <операторы>
}
```

Данный оператор последовательно проверяет на равенство переменной константам, стоящим после ключевого слова `case`. Если ни одна из констант не равна значению переменной, то выполняются операторы, находящиеся после слова `default`. Оператор `switch` имеет следующую особенность. Допустим, значение переменной равно значению константы1 и выполняются операторы, стоящие после первого ключевого слова `case`. После этого оператор принимает значение `true` и выполняет все операторы, следующие далее по остальным `case`. Во избежание такой ситуации следует использовать оператор `break` для перехода программы к следующему оператору после `switch`.

В листинге 4 представлен пример программирования условного оператора `switch`.

Листинг 4. Пример использования оператора `switch`.

```
#include <stdio.h>
int main()
{
    int x;
    printf("Введите число: ");
    scanf("%d", &x);
    switch(x)
```

```

{
    case 1 :
        printf("Введено число 1\n");
        break;
    case 2 :
        printf("Введено число 2\n");
        break;
    default :
        printf("Введено другое число\n");
}
char ch;
printf("Введите символ: ");
scanf("%c",&ch);
switch(ch)
{
    case 'a' :
        printf("Введен символ a\n");
        break;
    case 'b' :
        printf("Введен символ b\n");
        break;
    default :
        printf("Введен другой символ\n");
}
return 0;
}

```

Данный пример демонстрирует два разных варианта использования оператора switch. В первом случае выполняется анализ введенной цифры, во втором – анализ введенного символа. Следует отметить, что данный оператор может производить выбор только на основании равенства своего аргумента одному из перечисленных значений case, т.е. проверка выражений типа $x < 0$ в данном случае невозможна.

Задание на лабораторную работу

1. Составить блок-схему и написать программу работы с условным оператором if в соответствии с номером своего варианта.
2. Составить блок-схему и написать программу с использованием оператора switch в соответствии с номером своего варианта.
3. Сделать выводы о полученных результатах работы программ.

Варианты заданий

№	Оператор if	Оператор switch
1.	Даны 5 чисел. Вычислить сумму положительных среди них чисел.	С клавиатуры вводится порядковый номер месяца М [1;12]. Вывести приходящие на этот месяц праздники (например, если введен номер 1, выводится сообщение – ‘1 января Новый год’, ‘7 января – Рождество’ и т.п.).
2.	Написать программу проверки попадания введенного числа в диапазон от -2 до 2.	С клавиатуры вводится цифра [0;9]. Вывести на экран название цифры на английском языке (например, 2 – ‘two’ и т.п.).
3.	С клавиатуры последовательно вводятся числа K, L, M. Определить, является ли последовательность из этих чисел убывающей.	С клавиатуры вводится номер дня недели D [1;7]. Вывести название дня недели, соответствующее этому номеру.
4.	Даны 6 чисел. На сколько их сумма меньше или больше их произведения?	С клавиатуры вводится порядковый номер месяца М [1;12] текущего года. Определить количество дней в данном месяце.
5.	Написать программу проверки на положительность введенного числа	Написать программу замены введенного числа от 0 до 9 соответствующим символом
6.	С клавиатуры вводится год Y. Определить, будут ли проводиться в этом году зимние или летние олимпийские игры, или год не олимпийский.	С клавиатуры вводится порядковый номер цвета радуги R [1;7]. Вывести название цвета соответствующее этому номеру.
7.	С клавиатуры вводится целое число R. Определить кратно ли оно пяти или трем.	В старояпонском календаре принят двенадцатилетний цикл. Годы внутри цикла носили названия животных: крысы, коровы, тигра, кролика, дракона, змеи, лошади, овцы, обезьяны, петуха, собаки, и свиньи. Написать программу, которая по номеру года определяет его название в старояпонском календаре, если известно, что 1996 год был годом крысы - началом очередного цикла.
8.	Даны четыре числа. Найти разность между наибольшим и наименьшим среди них.	С клавиатуры вводится два вещественных числа X и Y (не равные нулю). Арифметические действия над ними пронумерованы следующим образом: 1- сложение, 2- вычитание; 3- умножение; 4- деление. Составить программу, которая по введенному

		номеру операции выполнит то или иное действие над числами.
9.	С клавиатуры вводится возраст человека V [1,100]. Определите стадию жизни человека по его возрасту, если 0..1 – Младенчество, 2..7 – Детство, 8..15 – Отрочество, 16..25 – Юность, 26..65 – Зрелость, 66..100 – Старость.	С клавиатуры вводится порядковый номер ноты T [1;7] Вывести название ноты, соответствующее этому номеру.
10.	Даны 6 чисел. Определить сколько среди них отрицательных и сколько положительных.	С клавиатуры вводится час суток T [1;24]. Определить время суток, по введенному часу, если считать: ночь с 22.00 до 5.00; утро с 6.00 до 11.00; день с 12.00 до 16.00; вечер с 17.00 до 21.00.
11.	С клавиатуры вводится целое трёхзначное число T . Определить, является ли оно палиндромом (т.е. читается ли число одинаково слева направо и справа налево).	Написать программу перевода введенного символа от А до F в нижний регистр, а все другие символы заменять на х
12.	Написать программу вычисления суммы модулей двух введенных чисел	С клавиатуры вводится номер дня недели D [1;7]. Вывести расписание занятий вашей группы в этот день или сообщение о выходном дне.
13.	С клавиатуры вводится время телефонных разговоров T (в минутах) за один месяц. Вычислить плату за пользование телефоном, если услуги телефонной сети оплачиваются по следующему правилу: 30 минут разговоров в месяц оплачиваются как абонентская плата в 100 рублей; разговоры сверх нормы оплачиваются из расчета 0,30 рублей за минуту.	С клавиатуры вводится целое число F [1;9]. Вывести на экран это число римскими цифрами.
14.	Даны 4 числа. Все отрицательные среди них числа заменить на 0.	С клавиатуры вводится оценка, полученная студентом – $Ос$. Выдать сообщение 'Умница', если $Ос=5$, 'Хорошо' – $Ос=4$, 'Лентяй' – $Ос=3$ или $Ос=2$.
15.	Даны 4 числа. Определить порядковый номер наименьшего среди них.	С клавиатуры вводится целое число N [1; 5]. Вывести фразу «Мы сдали N экзаменов», учитывая, что при некоторых значениях N слово «экзаменов» необходимо заменить словом «экзамен» или «экзамена».
16.	Даны 3 числа K , M и N . Поменять их значения местами таким образом, чтобы $K < M < N$.	С клавиатуры вводится порядковый номер пальца на руке N [1;5]. Вывести название пальца соответствующее этому номеру (например, если введено

		число 5, выводится сообщение – ‘мизинец’).
17.	Даны 3 числа. Поменять местами большее и меньшее из этих чисел.	С клавиатуры вводится первая буква названия падежа (И - именительный, Р - родительный, Д - дательный, В - винительный, Т - творительный, П - предложный). Вывести слово студент в заданном падеже единственного числа.

Содержание отчета

1. Титульный лист с названием лабораторной работы, фамилией студента и номером группы.
2. Цель работы.
3. Номер варианта и задание по варианту.
4. Блок-схемы алгоритмов.
5. Текст программ.
6. Результаты действия программ.
7. Выводы о полученных результатах работы программ.