Лабораторная работа №4

ОПЕРАТОРЫ ЦИКЛОВ ЯЗЫКА С

Цель работы: изучить особенности использования операторов цикла while, for и do while.

Теоретические сведения

Часто при создании программ на ЭВМ требуется много раз выполнить одну и ту же группу операторов. Например, для вычисления суммы ряда длиной N или перебора элементов массива с целью определения наибольшего или наименьшего значения и т.п. Во всех этих случаях необходим инструмент для реализации повторяющихся операций и таким инструментом являются операторы цикла.

Оператор цикла while

С помощью данного оператора реализуется цикл, который выполняется до тех пор, пока истинно условие цикла. Синтаксис данного оператора следующий:

```
while(<условие>)
{
 <тело цикла>
}
```

Приведем пример реализации данного цикла, в котором выполняется суммирование элементов ряда $S = \sum_{i=0}^{\infty} i$ пока $S < N$:

```
int N=20, i = 0;
long S = 0L;
while(S < N)
{
 S=S+i;
 i++;
}
```

В данном примере реализуется цикл while с условием i < N. Так как начальное значение переменной i=0, а N=20, то условие истинно и выполняется тело цикла, в котором осуществляется суммирование переменной i и увеличение ее на 1. Очевидно, что на 20 итерации значение i=20, условие станет ложным и цикл будет завершен. Продемонстрируем гибкость языка С, изменив данный пример следующим образом:

```
int N=20, i = 0;
long S = 0L;
while((S=S+i++) < N);
```

В данном случае при проверке условия сначала выполняются операторы, стоящие в скобках, где и осуществляется суммирование элементов ряда и только, затем, проверяется условие. Результат выполнения обоих вариантов программ одинаковый и S=21. Однако последняя конструкция бывает удобной при реализации опроса клавиатуры, например, с помощью функции scanf():

```
int num;
while(scanf("%d",&mun) == 1)
{
 printf("Вы ввели значение %d\n",num);
}
```

Данный цикл будет работать, пока пользователь вводит целочисленные значения и останавливается, если введена буква или вещественное число. Следует отметить, что цикл while можно принудительно завершить даже при истинном условии цикла. Это достигается путем использования оператора break. Перепишем предыдущий пример так, чтобы цикл завершался, если пользователь введет число 0.

```
int num;
while(scanf("%d",&mun) == 1)
{
   if(num == 0) break;
 printf("Вы ввели значение %d\n",num);
}
```

Цикл завершается сразу после использования оператора break, т.е. в приведенном примере, при вводе с клавиатуры нуля функция printf() выполняться не будет и программа перейдет на следующий оператор после while. Того же результата можно добиться, если использовать составное условие в цикле:

```
int num;
while(scanf("%d",&mun) == 1 && num != 0)
{
 printf("Вы ввели значение %d\n",num);
}
```

Таким образом, в качестве условия возможны такие же конструкции, что и в операторе if.

Оператор цикла for

Работа оператора цикла for подобна оператору while с той лишь разницей, что оператор for подразумевает изменение значения некоторой переменной и проверки ее на истинность. Работа данного оператора продолжается до тех пор, пока истинно условие цикла. Синтаксис оператора for следующий:

```
for(<инициализация счетчика>;<условие>;<изменение значения
счетчика>)
{
 <тело цикла>
}
```

Рассмотрим особенность реализации данного оператора на примере вывода таблицы кодов ASCII символов.

```
char ch;
for(ch = 'a'; ch <= 'z'; ch++)
      printf("Значение ASCII для %c - %d.\n",ch,ch);
```

В данном примере в качестве счетчика цикла выступает переменная ch, которая инициализируется символом ‘a’. Это означает, что в переменную ch заносится число 97 – код символа ‘a’. Именно так символы представляются в памяти компьютера. Код символа ‘z’ – 122, и все малые буквы латинского алфавита имеют коды в диапазоне [97; 122]. Поэтому, увеличивая значение ch на единицу, получаем код следующей буквы, которая выводится с помощью функции printf(). Учитывая все вышесказанное, этот же пример можно записать следующим образом:

```
for(char ch = 97; ch <= 122; ch++)
      printf("Значение ASCII для %c - %d.\n",ch,ch);
```

Здесь следует отметить, что переменная ch объявлена внутри оператора for. Это особенность языка С - возможность объявлять переменные в любом месте программы.

Существует много особенностей реализации данного оператора, отметим основные из них, которые могут заметно повысить скорость написания программ. Следующим примером продемонстрируем особенности изменения значения счетчика цикла.

```
int line_cnt = 1;
double debet;
for(debet = 100.0; debet < 150.0; debet = debet*1.1,
line_cnt++)
 printf("%d. Ваш долг теперь равен %.2f.\n",line_cnt, debet);
```

Следующий фрагмент программы демонстрирует возможность программирования сложного условия внутри цикла.

```
int exit = 1;
for(int num = 0;num < 100 && !exit; num += 1)
{
 scanf("%d",&mov);
 if(mov == 0) exit = 0;
 printf("Произведение num*mov = %d.\n",num*mov);
}
```

Оператор for с одним условием:

```
int i=0;
for(;i < 100;) i++;
```

и без условия

```
int i=0;
for(;;;) {i++; if(i > 100) break;}
```

В последнем примере оператор break служит для выхода из цикла for, т.к. он будет работать «вечно» не имея никаких условий.

Оператор цикла do while

Все представленные выше операторы циклов, так или иначе, проверяют условие перед выполнением цикла, благодаря чему существует вероятность, что операторы внутри цикла никогда не будут выполнены. Такие циклы называют циклы с предусловием. Однако бывают ситуации, когда целесообразно выполнять проверку условия после того, как будут выполнены операторы, стоящие внутри цикла. Это достигается путем использования операторов do while, которые реализуют цикл с постусловием. Следующий пример демонстрирует реализацию такого цикла.

```
const int secret_code = 13;
int code_ent;
do
{
 printf("Введите секретный код: ");
  scanf("%d",&code_ent);
}
while(code_ent != secret_code);
```

Из приведенного примера видно, что цикл с постусловием работает до тех пор, пока истинно условие, т.е. в данном случае пока значение введенного кода будет отличаться от значения секретного кода. Также следует обратить внимание на то, что после ключевого слова while должна стоять точка с запятой. При реализации данного цикла можно использовать составные условия, подобно циклу while, а также принудительно выходить из цикла с помощью оператора break.

Программирование вложенных циклов

Все рассмотренные выше операторы циклов допускают использование любых других операторов языка С внутри цикла, в том числе и операторов цикла. Это значит, что внутри одного цикла может находиться другой, что приводит к реализации вложенных циклов. Вложенные циклы необходимы для решения большого числа задач, например, вычисления двойных, тройных и т.д. сумм, просмотр элементов двумерного массива и многих других задач. В качестве примера вложенных циклов рассмотрим задачу вычисления суммы двойного ряда $S = \sum_{i=0}^{N} \sum_{j=0}^{M} i * j$:

```
long S = 0L;
int M = 10, N = 5;
for(int i = 0; i <= N;i++)
{
 for(int j = 0;j <= M;j++)
      S += i*j;
}
```

Того же результата можно добиться и с помощью оператора цикла while.

Задание на лабораторную работу

1. Написать программу работы с операторами циклов while и for в соответствии с номером своего варианта.
2. Написать программу с использованием оператора цикла do while в соответствии с номером своего варианта.
3. Сделать выводы о полученных результатах работы программ.

Варианты заданий

Вариант	Операторы циклов while и for	Оператор цикла do while
1	Вычислить $\sum_{i=1}^{50} 1/i^2$ с использованием оператора for	Написать программу ввода произвольных чисел до тех пор, пока не будет введено число 0
2	Вычислить $f(x) = kx + b$, при $x = 1,2,...,100$ с использованием оператора while	Написать программу ввода произвольных символов до тех пор, пока не будет введен символ q
3	Вычислить $\sum_{i=1}^{50} \sum_{j=1}^{30} i + j$ с помощью вложенных циклов for	Написать программу подсчета суммы 10 чисел, вводимых с клавиатуры

4	Вычислить $S=\sum_{i=1}^{\infty} i$ пока $S<50$ с помощью цикла while	Написать программу вычисления произведения 5 чисел, введенных с клавиатуры
5	Вычислить $S=\sum_{i=1}^{\infty} i^2$ пока $S<100$ с помощью цикла for	Написать программу вычисления модулей введенных чисел до тех пор, пока пользователь не введет 0
6	Вычислить $\sum_{i=1}^{50}\sum_{j=1}^{10} 1/(i+j)$ с помощью вложенных циклов while	Написать программу определения знака введенных чисел до тех пор, пока пользователь не введет 0
7	Вычислить $f(x)=x^2+b$, при $x=-10,-9,...,10$ с использованием оператора for	Написать программу определения минимального введенного числа из 10 чисел
8	Вычислить $\sum_{i=-10}^{10} 1/i^3$, $i\neq 0$ с использованием оператора for	Написать программу определения максимального введенного числа из 5 чисел
9	Вычислить $\sum_{i=-10}^{20}\sum_{j=0}^{10} 1/(i+j)^2$, $i+j\neq 0$ с помощью вложенных циклов for	Написать программу определения минимального среди положительных введенных 10 чисел
10	Вычислить $f(x)=1/x$, $x\neq 0$ при $x=-10,-9,...,10$ с использованием оператора for	Написать программу определения максимального среди отрицательных введенных 7 чисел

Содержание отчета

1. Титульный лист с названием лабораторной работы, номером варианта, фамилией студента и группы.
2. Текст программ.
3. Результаты действия программ.
4. Выводы о полученных результатах работы программ.

Контрольные вопросы

1. В чем отличия между операторами while и do while?
2. Дайте понятие вложенных циклов?
3. Что такое цикл с предусловием?
4. Что такое цикл с постусловием?
5. Условие остановки цикла while?
6. Для каких целей используются циклы в программировании?
7. Перечислите операторы циклов в языке С.