Лекция 4

Циклы

Циклы

- ***Циклом*** называется блок кода, который для решения задачи требуется повторить несколько раз.

  Каждый цикл состоит из
- блока проверки условия повторения цикла
- тела цикла
-

  Цикл выполняется до тех пор, пока блок проверки условия возвращает истинное значение.
  Тело цикла содержит последовательность операций, которая выполняется в случае истинного условия повторения цикла. После выполнения последней операции тела цикла снова выполняется операция проверки условия повторения цикла. Если это условие не выполняется, то будет выполнена операция, стоящая непосредственно после цикла в коде программы.

Циклы

- В языке Си следующие виды циклов:
- while — цикл с предусловием;
- do…while — цикл с постусловием;
- for — параметрический цикл (цикл с заданным числом повторений).

Параметрический цикл for

- Общая форма записи

- for (Инициализация; Условие; Модификация)
  {
    БлокОпераций;
  }

Параметрический цикл for

- for — параметрический цикл (цикл с фиксированным числом повторений). Для организации такого цикла необходимо осуществить три операции:
- **Инициализация** - присваивание параметру цикла начального значения;
- **Условие** - проверка условия повторения цикла, чаще всего - сравнение величины параметра с некоторым граничным значением;
- **Модификация** - изменение значения параметра для следующего прохождения тела цикла.

Параметрический цикл for

- Эти три операции записываются в скобках и разделяются точкой с запятой **;**;. Как правило, параметром цикла является целочисленная переменная. **Инициализация** параметра осуществляется только один раз — когда цикл for начинает выполняться.
  Проверка **Условия** повторения цикла осуществляется перед каждым возможным выполнением тела цикла. Когда выражение, проверяющее **Условие** становится ложным (равным нулю), цикл завершается. **Модификация** параметра осуществляется в конце каждого выполнения тела цикла. Параметр может как увеличиваться, так и уменьшаться.

Пример. Посчитать сумму чисел от 1 до k

```
int k = 5;  // объявляем целую переменную k
int i = 1;
int sum = 0; // начальное значение суммы равно 0
int main()
{
for(i=1; i<=k; i++) // цикл для переменной i от 1 до k с шагом 1
  {
    sum = sum + i; // добавляем значение i к сумме
  }
  printf("sum = %03d\n", sum);// sum = 15
  return 0
}
```

Параметрический цикл for

- В записи цикла for можно опустить одно или несколько выражений, но нельзя опускать точку с запятой, разделяющие три составляющие цикла.
- int i = 1;
- …….
- void main(void)
- {
- for(; i<=k; i++) // цикл для переменной i от 1 до k с шагом 1
- {
- ……..
- }
- ……..
- }

Параметрический цикл for

- **Вложенные циклы**
- В Си допускаются вложенные циклы, то есть когда один цикл находится внутри другого:

- ```
  for (i = 0; i<n; i++)  // внешний цикл - Цикл1
  {
    for (j = 0; j<n; j++)   // вложенный цикл - Цикл2
    {
      ;        // блок операций Цикла2
    }
    // блок операций Цикла1;
  }
  ```
  ```

Пример. Вывести числа от 0 до 99, по 10 в каждой строке

```
int i,j;

void main(void) {
  for( i=0; i<10; i++) // цикл для десятков
  {
    for (j = 0; j < 10; j++) // цикл для единиц
    {
      printf("sum = %02d\n", i * 10 + j);
    }
  }
}
```

Пример. Вывести числа от 0 до 99, по 10 в каждой строке

```
 0  1  2  3  4  5  6  7  8  9
10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27 28 29
30 31 32 33 34 35 36 37 38 39
40 41 42 43 44 45 46 47 48 49
50 51 52 53 54 55 56 57 58 59
60 61 62 63 64 65 66 67 68 69
70 71 72 73 74 75 76 77 78 79
80 81 82 83 84 85 86 87 88 89
90 91 92 93 94 95 96 97 98 99
```

Операторы прерывания и продолжения цикла break и continue

- В теле любого цикла можно использовать операторы прерывания цикла - break и продолжения цикла - continue.

  Оператор break позволяет выйти из цикла, не завершая его.
  Оператор continue позволяет пропустить часть операторов тела цикла и начать новую итерацию.

Пример. Вывести числа от 0 до 99 ниже главной диагонали

```
int i,j;
void main(void) {
  for(int i=0; i<10; i++) // цикл для десятков
  {
    for (int j = 0; j < 10; j++) // цикл для единиц
    {
      if (j > i) // если число единиц больше числа десятков в числе
        break; // выходим из вложенного цикла и переходим к новой строке
      printf("sum = %02d\n", i * 10 + j);
    }
  }
}
```

Пример. Вывести числа от 0 до 99 ниже главной диагонали

```
 0
10 11
20 21 22
30 31 32 33
40 41 42 43 44
50 51 52 53 54 55
60 61 62 63 64 65 66
70 71 72 73 74 75 76 77
80 81 82 83 84 85 86 87 88
90 91 92 93 94 95 96 97 98 99
```

Вывести числа от 0 до 99 исключая числа, оканчивающиеся на 5 или 8

```
int i,j;

void main(void) {
  for(int i=0; i<10; i++) // цикл для десятков
  {
    for (int j = 0; j < 10; j++) // цикл для единиц
    {
      if ((j == 5) || (j == 8)) // если число единиц в числе равно 5 или 8,
        continue;              // переходим к следующей итерации цикла
      printf("sum = %02d\n", i * 10 + j);
    }
  }
}
```

Вывести числа от 0 до 99 исключая числа, оканчивающиеся на 5 или 8

```
 0  1  2  3  4  6  7  9
10 11 12 13 14 16 17 19
20 21 22 23 24 26 27 29
30 31 32 33 34 36 37 39
40 41 42 43 44 46 47 49
50 51 52 53 54 56 57 59
60 61 62 63 64 66 67 69
70 71 72 73 74 76 77 79
80 81 82 83 84 86 87 89
90 91 92 93 94 96 97 99
```

Цикл с предусловием while

- Общая форма записи
  while (Условие)
  {
  БлокОпераций;
  }

  Если **Условие** выполняется (выражение, проверяющее **Условие**, не равно нулю), то выполняется **БлокОпераций**, заключенный в фигурные скобки, затем **Условие** проверяется снова.
  Последовательность действий, состоящая из проверки **Условия** и выполнения **БлокаОпераций**, повторяется до тех пор, пока выражение, проверяющее **Условие**, не станет ложным (равным нулю). При этом происходит выход из цикла, и производится выполнение операции, стоящей после оператора цикла.

Пример. Вывести на дисплей все числа от 6 до 1

```
int i = 1;
void main(void)
{
while (i > 0)
    {
        printf("sum = %03d\n", i);
        i--;
    }
}
```

Пример. Посчитать сумму чисел от 1 до k

```
int k = 5;  // объявляем целую переменную k
int i = 1;
int sum = 0; // начальное значение суммы равно 0
void main(void)
{
while (i <= k)     // пока i меньше или равно k
  {
    sum = sum + i; // добавляем значение i к сумме
    i++;           // увеличиваем i на 1
  }
  printf("sum = %03d\n", sum);// sum = 15
}
```

Цикл с предусловием while

- При построении цикла while, в него необходимо включить конструкции, изменяющие величину проверяемого выражения так, чтобы в конце концов оно стало ложным (равным нулю). Иначе выполнение цикла будет осуществляться бесконечно (бесконечный цикл).
- while (1)
  {
  БлокОпераций;
  }

Цикл с постусловием do...while

- Общая форма записи
- do {
  БлокОпераций;
  } while (Условие);

  Цикл do...while — это цикл с постусловием, где истинность выражения, проверяющего **Условие** проверяется после выполнения **Блока Операций**, заключенного в фигурные скобки. Тело цикла выполняется до тех пор, пока выражение, проверяющее **Условие**, не станет ложным, то есть тело цикла с постусловием выполнится хотя бы один раз.

Пример. Дано натуральное число. Определить количество цифр 7 в нем.

```
int number;
int t;
int k;
void main(void)
{
    t = number; // сделать копию из number
    k = 0;
   // цикл вычисления k
      do
      {
        if (t % 10 == 7) k++;
        t = t / 10;
      } while (t > 0);
      printf("sum = %03d\n", k);
}
```

Пример. Дано натуральное число. Определить, есть ли последовательность его цифр при просмотре слева направо упорядоченной по убыванию.

```
int number = 9621;
int t1, t2;
bool f;
void main(void)
{
    // Вычисление
    t1 = number % 10;
    f = true;
    // цикл do-while
    do
    {
      t2 = t1;
      number = number / 10;
      t1 = number % 10;
      if (t1 < t2)
      {
        f = false;
        break; // последовательность не убывающая
      }
    } while (number > 10);

    if (f == true)
      printf("Decreasing.");
    else
      printf("Not decreasing.");
}
```