**Цикл while**

В языке C++ существует три вида циклов: цикл while c предусловием, цикл while с постусловием, цикл for.

**Цикл while ("пока") с предусловием**

Цикл while с предусловием позволяет выполнить одну и ту же последовательность действий пока проверяемое условие истинно. При этом условие записывается до тела цикла и проверяется до выполнения тела цикла.

При выполнении цикла while сначала проверяется условие. Если оно ложно, то цикл не выполняется и управление передается на следующую инструкцию после тела цикла while. Если условие истинно, то выполняется инструкция, после чего условие проверяется снова и снова выполняется инструкция. Так продолжается до тех пор,**пока** условие будет истинно. Как только условие станет ложно, работа цикла завершится и управление передастся следующей инструкции после цикла.

Синтаксис цикла while ("пока") c предусловием такой:

     while (условие)  
     {  
         блок инструкций  
     }

Следующий фрагмент программы напечатает на экран квадраты всех целых чисел от 1 до 10:

     int i=1;  
     while (i<=10)  
     {  
         cout<<i\*i<<endl;  
         ++i;  
     }

В этом примере переменная i внутри цикла изменяется от 1 до 10. Такая переменная, значение которой меняется с каждым новым проходом цикла, называется*счетчиком*. Заметим, что после выполнения этого фрагмента значение переменной i будет равно 11, поскольку именно при i==11 условие i<=10 впервые перестанет выполняться.

В следующем примере цикл используется для того, чтобы найти количество знаков в десятичной записи целочисленной переменной i.

     int Ndigits=0;  
     while(n!=0)  
     {  
         Ndigits=Ndigits+1;  
         n=n/10;  
     }

Внутри цикла значение переменной n уменьшается в 10 раз до тех пор, пока она не станет равна 0. Уменьшение целочисленной переменной в 10 раз (с использованием целочисленного деления) эквивалентно отбрасыванию последней цифры этой переменной.

**Цикл while ("пока") с постусловием**

Цикл "пока" с постусловием отличается от цикла с предусловием тем, что сначала выполняется блок цикла, а потом проверяется условие. Если условие истинно, то цикл будет выполнен еще раз, и так до тех пор, пока условие будет истинно. Синтаксис цикла с постусловием такой (обратите внимание на обязательную точку с запятой после условия):

     do  
     {  
         Блок инструкций  
     }  
     while (условие);

Поскольку условие проверяется после выполнения тела цикла, то блок цикла с постусловием всегда будет выполнен хотя бы один раз, независимо от истинности условия. Это может привести к ошибкам, поэтому использовать цикл while с постусловием следует только тогда, когда это действительно упрощает алгоритм.

**Упражнения**

Во всех задачах этого листочка основная (содержательная) часть программы должна быть оформлена в виде функции. Функция main должна считывать значения входных данных, вызывать основную функцию, решающую задачу, выводить результат на экран.

1. (A) Напечатайте все точные квадраты натуральных чисел, не превосходящие данного числа n. (например, при вводе 50 программа должна вывести 1 4 9 16 25 36 49).
2. (B) Дано натуральное число n. Определите, является ли оно степенью числа 2, и выведите слово YES, если является и слово NO, если не является.
3. (C) Для данного натурального числа n определите такое наименьшее целое k, что 2k≥n. Например, при вводе числа 7 программа должна вывести 3.
4. (D YES) В первый день спортсмен пробежал x километров, а затем он каждый день увеличивал пробег на 10% от предыдущего значения. По данному числу y определите номер дня, на который пробег спортсмена составит не менее y километров. Например, при вводе 10 20 программа должна вывести 9. x и y – действительные числа, ответ – целое число.
5. (E) В первый день спортсмент пробежал x километров, а затем он каждый день увеличивал пробег на 10% от предыдущего значения. По данному числу y определите номер дня, на который суммарный пробег спортсмена составит не менее y километров. Например, при вводе 10 100 программа должна вывести 8.
6. (F) Дано натуральное число n. Напишите функцию int SumOfDigits (int n), вычисляющую сумму цифр числа n. Выведите сумму цифр числа n.
7. (G) Дано натуральное число n. Напишите функцию int NumberOfZeroes (int n), определяющую количество нулей среди всех цифр числа n. Выведите результат.
8. (H) Дано натуральное число n. Напишите функцию int MinDigit (int n) и int MaxDigit (int n), определяющую наименьшую и наибольшую цифры данного числа. Выведите наименьшую и наибольшую цифры данного числа (например, при вводе 179 программа выводит 1 9).
9. (I) Вводится последовательность целых чисел до тех пор, пока не будет введено число 0. После ввода числа 0 программа должна завершить свою работу и вывести сумму введенных чисел.
10. (J) По данному натуральному числу n найдите сумму чисел 1+1/1!+1/2!+1/3!+...+1/n!. Количество действий должно быть пропорционально n. Напишите программу, которая считывает значение n и выводит результат в виде действительного числа. К чему будет стремиться эта сумма при росте числа n?
11. (K) По данному числу n выведите n-e число Фибоначчи. Использовать рекурсию нельзя.
12. (L) Напишите функцию int reverse(int n), которая переставляет цифры числа в обратном порядке (например, reverse(179)==971). Напишите программу, которая по данному натуральному n печатает его цифры в обратном порядке.
13. (M) Назовем число палиндромом, если оно не меняется при перестановке его цифр в обратном порядке. Напишите функцию bool IsPalindrome (int n), проверяющую по данному числу n, является ли оно палиндромом. Напишите программу, которая по заданному числу K выводит количество натуральных палиндромов, не превосходящих K. Например, при вводе 1 программа выводит 1, а при вводе 100 программа выводит 18.