

Лабораторная работа 2. Операторы Цикла.

Как организовать цикл?

Цикл –это многократное выполнение одинаковых действий. Доказано, что любой алгоритм может быть записан с помощью трёх алгоритмических конструкций: циклов, условных операторов и последовательного выполнения команд (линейных алгоритмов).

Простейший цикл, который 10 раз выводит на экран слово «привет», на псевдокоде записывается так:

сделай 10 раз вывод "привет"

Подумаем, как можно организовать такой цикл. Вы знаете, что программа после запуска выполняется автоматически. И при этом на каждом шаге нужно знать, сколько раз уже выполнен цикл и сколько ещё осталось выполнить. Для этого необходимо использовать ячейку памяти, в которой будет запоминаться количество выполненных шагов цикла (счётчик шагов). Сначала можно записать в неё ноль (ни одного шага не сделано), а после каждого шага цикла увеличивать значение ячейки на единицу. На псевдокоде алгоритм можно записать так (здесь и далее операции, входящие в тело цикла, выделяются отступами):

счётчик = 0

пока счётчик < 10

{вывод "привет" "увеличить счётчик на 1}

Возможен и другой вариант: сразу записать в счётчик нужное количество шагов, и после каждого шага цикла уменьшать счётчик на 1. Тогда цикл должен закончиться при нулевом значении счётчика:

счётчик = 10

пока счётчик > 0

{вывод "привет"

уменьшить счётчик на 1}

Этот вариант несколько лучше, чем предыдущий, поскольку счётчик сравнивается с нулём, а такое сравнение выполняется в процессоре автоматически. В этих примерах мы использовали цикл с условием, который выполняется до тех пор, пока некоторое условие не становится ложно.

Циклы с условием.

Рассмотрим следующую задачу: *определить количество цифр в десятичной записи целого положительного числа. Будем предполагать, что исходное число записано в переменную n целого типа.*

Сначала составим алгоритм решения этой задачи. Чтобы считать что-то в программе, нужно использовать переменную, которую называют счётчиком. Для подсчёта количества цифр необходимо как-то отсекают эти цифры по одной, с начала или с конца, каждый раз увеличивая счётчик. Начальное значение счётчика должно быть равно нулю, так как до выполнения алгоритма ещё не найдено ни одной цифры. Для отсечения первой цифры необходимо заранее знать, сколько цифр в десятичной записи числа, то есть нужно заранее решить ту задачу, которую мы решаем. Следовательно, этот метод не подходит. Отсечь последнюю цифру проще – достаточно разделить число нацело на 10 (поскольку речь идет о десятичной системе). Операции отсечения и увеличения счётчика нужно выполнять столько раз, сколько цифр в числе. Как же «поймать» момент, когда цифры кончатся? Не сложно понять, что в этом случае результат очередного деления на 10 будет равен нулю, это и говорит о том, что отброшена последняя оставшаяся цифра.

Псевдокод выглядит так:

счётчик = 0

пока $n > 0$

{отсечь последнюю цифру n

увеличить счётчик на 1}

Циклы на языке C

```
count = 0;
```

```
while ( n > 0 ) {
```

```
    n = n / 10;
```

```
    count ++;
```

```
}
```

Здесь целочисленная переменная - счётчик имеет имя count. Слово while переводится с английского как «пока», за ним в скобках записывается условие работы цикла (в данном случае – «пока $n > 0$ »). Фигурные скобки ограничивают составной оператор. Если в теле цикла нужно выполнить только один оператор, эти скобки можно не ставить. Напомним, что операция деления для целых чисел всегда даёт целое число (остаток отбрасывается). Обратите внимание, что проверка условия выполняется в начале очередного шага цикла. Такой цикл называется циклом с

предусловием (то есть с предварительной проверкой условия) или циклом «пока». Если в начальный момент значение переменной *n* будет нулевой или отрицательное, цикл не выполнится ни одного раза.

Вложенные циклы

В более сложных задачах часто бывает так, что на каждом шаге цикла нужно выполнять обработку данных, которая также представляет собой циклический алгоритм. В этом случае получается конструкция «цикл в цикле» или «вложенный цикл».

Предположим, что нужно найти все простые числа в интервале от 2 до 1000. Простейший (но не самый быстрый) алгоритм решения такой задачи на псевдокоде выглядит так:

сделать для n от 1 до 1000

если число n простое, то вывод n

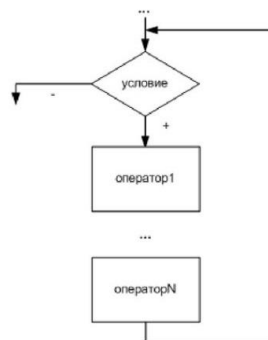
Как же определить, что число простое? Как известно, простое число делится только на 1 и само на себя. Если число *n* не имеет делителей в диапазоне от 2 до *n*-1, то оно простое, а если хотя бы один делитель в этом интервале найден, то составное. Чтобы проверить делимость числа *n* на некоторое число *k*, нужно взять остаток от деления *n* на *k*. Если этот остаток равен нулю, то *n* делится на *k*. Таким образом, программу можно записать так (здесь *n*, *k* и *count* – целочисленные переменные, *count* обозначает счётчик делителей):

```
for (n = 2; n <= 1000; n ++){  
    count= 0;  
    for (k = 2; k < n; k ++ )  
        if ( n % k == 0 ) count ++;  
    if ( count == 0 ) cout << n << endl;  
}
```

Цикл с предусловием

`while (условие)
оператор;`

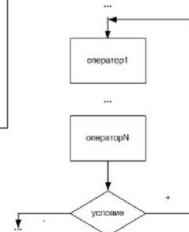
Сталкиваясь с оператором `while`, программа будет оценивать условие цикла. Если условие истинно, ваша программа выполняет операторы цикла `while`. После выполнения последнего оператора цикла программа снова проводит проверку условия. Если условие истинно, программа повторит этот процесс, выполнит операторы, а затем повторит проверку условия. Если условие оценивается как ложь, программа продолжит свое выполнение с первого оператора, который следует за оператором `while`.



Цикл постусловием

`do {
оператор;
} while (условие);`

- Когда программа встречает оператор `do while`, она сразу же выполняет операторы, содержащиеся в цикле. Затем программа исследует условие цикла. Если условие истинно, программа повторяет операторы цикла и процесс продолжается. Если условие цикла становится ложным, программа продолжает свое выполнение с первого оператора, следующего за оператором `do while`.



Задачи.

Задача 1.

По заданному натуральному n , вычислить:

$$1 - \frac{1}{3} + \frac{1}{5} - \dots \pm \frac{1}{2n+1}$$

Задача 2.

В последовательности a_1, a_2, \dots, a_n найдите номер самого большого числа.

Входные данные

На вход программе сначала подается натуральное число $n \leq 10^6$. В следующей строке следуют n целых чисел, по модулю не превосходящих 30000, — сами члены последовательности.

Выходные данные

Выведите номер максимального числа. Если таких чисел несколько, то выведите номер последнего из них. Нумерация чисел начинается с единицы.

Массив в программе не использовать.

Задача 3.

По заданным вещественному значению x и целому значению n вычислите x^n . Для решения задачи используйте последовательное домножение результата на x .

Нельзя использовать встроенную операцию возведения в степень.

Входные данные

На вход программе подаются вещественное x , по модулю не превосходящее 10, и целое n , по модулю не превосходящее 20.

Выходные данные

Выведите значение x^n с точностью не меньше, чем 10^{-3} .

Задача 4.

Вкладчик положил на банковский счет n рублей. Каждый год на сумму вклада начисляется k процентов годовых (будем считать, что процент всегда округляется до целого числа рублей по формуле $[xk/100]$, где x — сумма вклада на начало года). Начисленные проценты добавляются к сумме вклада. Через сколько лет сумма вклада станет не менее m рублей?

Входные данные

На вход программе подаются три натуральных числа: $n \leq 10^6$, $k \leq 100$, $m \leq 1000n$.

Выходные данные

Выведите одно число — искомое количество лет.

Индивидуальные задание 1.

Разработать программу на языке программирования C++.

В диалоговом режиме ввести число N (В диапазоне от 1 до 1000). Программа должна вывести искомые числа в виде нескольких колонок, выровненных по правому краю. Все числа от 1 до N натуральные.

1.Распечатать все числа от 1 до N , у которых остатки от деления на число Z не превышают числа M .

2.Распечатать квадраты только 3-значных чисел от 1 до N .

3.Распечатать все числа от 1 до N , не оканчивающиеся на цифру 3.

4.Распечатать корни всех чисел от 1 до N , не имеющих однозначных делителей (не равных числу).

5.Распечатать все числа от 1 до N, у которых младшая цифра кратна текущему числу.

6.Распечатать все числа от 1 до N, имеющие делители 3, 4 и 7.

7.Распечатать квадраты всех нечетных чисел от 1 до N, кратных порядковому номеру текущего числа.

8.Распечатать все числа от 1 до N, у которых младшая цифра является делителем числа N.

9.Распечатать все числа от 1 до N, у которых нет двухзначных делителей (не равных числу).

10.Распечатать кубы всех нечетных чисел от 1 до N.

11.Распечатать все числа от 1 до N, у которых самый большой делитель (не равный числу) есть однозначное число.

12.Распечатать все числа от 1 до N, у которых первая и вторая цифра справа равны числу M.

13.Распечатать все числа от 1 до N, у которых есть хотя бы один двухзначный делитель (не равный числу).

14.Распечатать квадраты всех нечетных чисел от 1 до N, вторая цифра справа которых четна.

15.Распечатать все числа от 1 до N, у которых имеется делитель (не равный числу), кратный числу.

16.Распечатать все числа от 1 до N, у которых первая и вторая цифра справа не равны

17. Распечатать все числа от 1 до N, у которых первая и последняя цифра совпадают.

18. Распечатать все простые числа от 1 до N.

19. Распечатать кубы всех четных чисел от 1 до N.

20. Распечатать все числа от 1 до N, у которых сумма крайних цифр совпадает с суммой средних.

Индивидуальные задание 2.

Для функции $f(x) = xe^x + 2\sin x - \sqrt{|x^3 - x^2|}$ составить программу построения таблицы значений функции при изменении аргумента от $A = -5$ до $B = 5$ с шагом $H = 0,1$. Значения A , B и H объявить как константы в директивах `#define`. В каждой строке выводить значение аргумента и соответствующее ему значение функции в форматированном виде с 4 знаками после запятой. Кроме того, в конце таблицы нужно напечатать отдельной строкой значение, которое требуется вычислить в соответствии с индивидуальным вариантом.

1. Количество отрицательных значений функции, имеющих нечетную целую часть.
2. Среднее арифметическое таких значений функции, которые имеют в младшем разряде целой части цифру, большую 3.
3. Среднее арифметическое таких значений функции, которые имеют дробную часть, меньшую 0,5.
4. Количество значений функции, имеющих в младшем разряде целой части цифру, большую 3.
5. Среднее арифметическое таких значений функции, которые имеют четную целую часть.
6. Количество неотрицательных значений функции, имеющих четную целую часть.
7. Сумма таких значений функции, которые имеют дробную часть, большую 0,5.
8. Сумма таких значений функции, которые имеют целую часть, кратную Z .
9. Среднее арифметическое таких значений функции, которые имеют нечетную целую часть.
10. Количество положительных значений функции, которые имеют четную целую часть и дробную часть, меньшую 0,5.
11. Сумма неотрицательных значений функции, которые имеют нечетную целую часть и дробную часть, большую 0,2.
12. Среднее арифметическое таких значений функции, которые имеют нечетную целую часть и дробную часть, большую 0,5.
13. Сумма таких значений функции, целая часть которых двузначна.
14. Количество таких значений функции, целая часть которых по модулю меньше числа M .
15. Сумма таких значений функции, дробная часть которых больше 0,3 и меньше 0,7.

16. Среднее арифметическое таких значений функции, целая часть которых по модулю больше числа M .

17. Среднее арифметическое таких значений функции, дробная часть которых больше 0,2 и меньше 0,8.

18. Среднее геометрическое таких значений функции, целая часть которых по модулю больше числа M .

19. Среднее геометрическое таких значений функции, дробная часть которых больше 0,2 и меньше 0,8.

20. Среднее квадратичное таких значений функции, целая часть которых по модулю больше числа M .

21. Среднее квадратичное таких значений функции, дробная часть которых больше 0,2 и меньше 0,8.

Индивидуальное задание 3.

Разработать программу, которая по заданному числу $n > 0$, вычисляет значение числового ряда, согласно варианту. Вывести промежуточные расчеты в виде таблицы, содержащей n строк (в каждой отдельной строке текущий числитель и знаменатель, текущее значение дроби и суммы). Также необходимо найти наименьшее и наибольшее значение слагаемого. Выводить значения с точностью 5 знаков после запятой.

Пример консоли вывода программы:

```
N=5
X=1
Числитель      Знаменатель      Слагаемое      Сумма
1              1              1              1
1              6              0.16667        0.83333
1              120             0.0083333      0.84167
1              5040            0.00019841     0.84147
1              362880          2.7557e-06     0.84147
max=1
min=2.7557e-06
Для продолжения нажмите любую клавишу . . .
```


$$1. y = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots;$$

$$2. y = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots;$$

$$3. y = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots;$$

$$4. S = \frac{1}{(x)^2} + \frac{2}{(x)^3} + \frac{3}{(x)^4} + \frac{5}{(x)^5} + \frac{8}{(x)^6} + \dots;$$

$$5. S = \frac{1!}{13} - \frac{2!}{15} + \frac{3!}{28} - \frac{4!}{43} + \dots;$$

$$6. S = \frac{1!}{(x+1)^3} - \frac{2!}{(x+2)^3} + \frac{3!}{(x+3)^3} \dots;$$

$$7. S = \frac{1}{(2x-1)^2} + \frac{1}{(3x+2)^2} + \frac{1}{(4x-3)^2} + \dots;$$

$$8. S = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + \dots;$$

$$9. y = x - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots;$$

$$10. y = \frac{x}{a} - \frac{x^3}{\sqrt{a}} + \frac{x^5}{\sqrt{\sqrt{a}}} - \frac{x^7}{\sqrt{\sqrt{\sqrt{a}}}} + \dots;$$

$$11. y = 1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots;$$

$$12. y = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots;$$

$$13. y = 1 - \frac{3}{2} + \frac{3 \cdot 5}{2 \cdot 4} \cdot x^2 - \frac{3 \cdot 5 \cdot 7}{2 \cdot 4 \cdot 6} \cdot x^3 + \dots;$$

$$14. y = -(1+x)^2 + \frac{(1+x)^4}{2} - \frac{(1+x)^6}{4} + \dots;$$

$$15. P = \sum_{i=1}^N \frac{A}{(3i-1) \cdot (3i+2)};$$

$$16. Y = \sum_{k=1}^N \frac{1}{k \cdot (k+1) \cdot (k+2)};$$

$$17. Z = \sum_{k=1}^N \frac{1}{(2+3k) \cdot (5+3k) \cdot (8+3k)};$$

$$18. M = \sum_{k=1}^N \frac{(-1)^{k-1}}{k^2 + 9k + 1};$$

$$19. S = \frac{1}{2} + \frac{1}{5} + \frac{1}{10} + \dots + \frac{1}{k^2+1} + \dots$$

$$20. S = \frac{1}{1 \cdot 2 \cdot 3} - \frac{1}{2 \cdot 3 \cdot 4} + \frac{1}{3 \cdot 4 \cdot 5} - \dots$$

$$21. S = \sum_{k=1}^N \left(-\frac{2}{5}\right)^{k+1};$$

$$22. S = \sum_{k=1}^N \frac{1}{(2k-1) \cdot (2k+1)};$$

$$23. S = \frac{1}{3} - \frac{2}{5} + \frac{3}{7} - \dots + \frac{k}{2k+1} + \dots;$$

$$24. S = \sum_{k=1}^N \frac{2k-1}{2^k};$$

$$25. S = \sum_{k=1}^N \frac{2k+1}{\sqrt{k \cdot 2^k}};$$