

Лабораторная работа. Циклические операторы

1. Написать программу, которая выводит на экран температуры в градусах Цельсия от 0° до 100° с шагом 10° и их эквиваленты в градусах Фаренгейта.

Для перевода используется формула: $t_F = \frac{9}{5}t_C + 32$

а) циклом while

б) циклом do..while

в) циклом for вывести таблицу значений функции $y = \cos(x)$ в диапазоне от 0 до 2π с шагом $\pi/6$

2. Среди 7-ми введенных действительных чисел определить (массивы не использовать)

а) количество положительных чисел

б) сумму отрицательных чисел

в) наибольшее из введенных

3. Вводить целые числа в диалоге с пользователем до тех пор, пока он не откажется от ввода (хотя бы одно число он должен обязательно ввести). Вывести общее количество введенных чисел и количество среди них четных чисел.

4. а) дано целое число n . Вычислить и вывести сумму $S = \frac{1}{2} + \frac{1}{4} + \frac{1}{6} + \frac{1}{8} + \dots + \frac{1}{2n}$

б) дано маленькое положительное число (например $\varepsilon = 0.001$). Реализовать алгоритм приближенного вычисления бесконечной суммы. Нужное приближение считается полученным, если вычислена сумма нескольких первых слагаемых, и модуль следующего слагаемого меньше данного положительного числа.

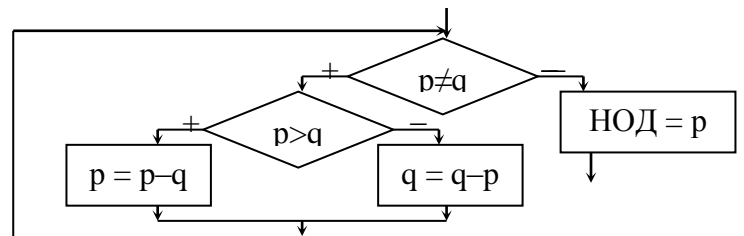
$$1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots \quad \left(\approx \frac{\pi}{4} \right) \text{ ответ для тестирования}$$

5. Вычислите наибольший общий делитель (НОД) двух целых чисел по алгоритму Евклида:

На экран выведите исходные числа и НОД для них.

Например: для $p=18$ и $q=24$ НОД(18, 24) = 6.

Правильно организуйте работу алгоритма для отрицательных чисел.



Используя в программе циклические операторы for решите следующие задачи

6. Дано вещественное число $B > 0$. Последовательность чисел образуется по правилу: $a_0 = 1$,

$a_{i+1} = a_i^2 + 1$. Распечатать все числа a_i такие, что $a_i < B$. Вывести количество напечатанных чисел.

7. Написать программу вывода на экран текстового изображения шахматной доски (белые клетки можно обозначить, например, пробелом или символом 'о', а черные – символом '*').

8. Написать программу, выводящую на экран таблицу умножения чисел от 1 до 10

9. Вводятся данные о координатах точек на плоскости (x_i, y_i) , $i=1, 2, \dots, N$, где N заранее указывает пользователь. Программа вычисляет и печатает количество точек в каждом из квадрантов (углов) координатной плоскости.

10. Найти периметр n -угольника, вершины которого имеют соответственно координаты $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$. Число n и координаты вводятся пользователем

11. Написать программу, которая позволяет ввести n чисел. Программа выводит сообщение о том имелась ли во введенной последовательности чисел хоть одна пара находящихся рядом одинаковых чисел.

Дополнительные задания

(используя любые циклы)

12. Написать программу приближенного вычисления интеграла функции $f(x)$ методом прямоугольников. Приближенное значение интеграла определяется как сумма площадей прямоугольников, нижними сторонами которых являются длины отрезков интегрирования, а длины боковых сторон соответствуют значениям функции $f(x)$ в серединах

$$\int_0^{\pi} \sin(x) dx = 2$$

отрезков). Проверить работу программы на функции с известным ответом, например,

13. Задана функция и ее разложение в ряд. Вычислить разложение в ряд с заданной погрешностью ξ (малое положительное число) и сравнить с точным значением функции. Оценить требуемое для достижения заданной точности число итераций.

а)
$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + (-1)^n \frac{x^{2n}}{(2n)!} + \dots$$

б)
$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + (-1)^{n-1} \frac{x^{2n-1}}{(2n-1)!} + \dots$$

14. Известно время начала и окончания работы автобусного маршрута с одним автобусом на линии (например, 6:00 и 23:30), а также протяженность маршрута в один конец (в минутах) и время отдыха на конечных остановках. Вывести суточное расписание этого маршрута (моменты отправления с конечных пунктов) без учета времени на обед и пересмену.

15. Прямоугольник на плоскости $a \leq x \leq b$, $c \leq y \leq d$ задается своими габаритами a, b, c, d . Последовательно вводятся габариты N прямоугольников. В процессе ввода находить сумму площадей их пересечения, не запоминая самих габаритов.

