

Требования к программам

1. Программа должна получать все параметры в качестве аргументов командной строки.
2. Задачи оцениваются независимо в двух группах: задачи 1–6 и задачи 7–12.
3. Аргументы командной строки для задачи 1: нет аргументов
4. Аргументы командной строки для задач 2, 3:

- 1) x – вещественное число.

Например, запуск

```
./a.out 3.001
```

означает, что $x = 3.001$.

5. Аргументы командной строки для задач 4, 5:

- 1) a – вещественное число,
- 2) b – вещественное число,
- 3) c – вещественное число.

Например, запуск

```
./a.out 1 2 3
```

означает, что $a = 1$, $b = 2$, $c = 3$.

6. Аргументы командной строки для задач 6, 7, 10–12:

- 1) n – целое число.

7. Аргументы командной строки для задач 8, 9:

- 1) x – вещественное число,
- 2) n – целое число.

8. Решение задачи должно быть оформлено в виде функции.
9. В функции, реализующей решение задачи, запрещается что-либо выводить на экран.
10. Вывод результата работы в функции `main` для задач 1, 3, 6–10 должен производиться по формату:

```
printf ("%s : Task = %d Result = %e\n",  
        argv[0], task, res);
```

где

- `argv[0]` – первый аргумент командной строки (имя образа программы),
- `task` – номер задачи,

- `res` – возвращаемое значение функции, реализующей решение этой задачи,

11. Вывод результата работы в функции `main` для задачи 4 должен производиться по формату:

```
printf ("%s : Task = %d X1 = %e X2 = %e\n",
        argv[0], task, x1, x2);
```

где

- `argv[0]` – первый аргумент командной строки (имя образа программы),
- `task` – номер задачи,
- `x1, x2` – результаты работы (полученные значения).

12. Вывод результата работы в функции `main` для задачи 5 должен производиться по формату:

```
printf ("%s : Task = %d Num = %d X1 = %e X2 = %e\n",
        argv[0], task, num, x1, x2);
```

где

- `argv[0]` – первый аргумент командной строки (имя образа программы),
- `task` – номер задачи,
- `num` – возвращаемое значение функции, реализующей решение этой задачи,
- `x1, x2` – результаты работы (полученные значения).

13. Вывод результата работы в функции `main` для задач 2, 11, 12 должен производиться по формату:

```
printf ("%s : Task = %d Result = %d\n",
        argv[0], task, res);
```

где

- `argv[0]` – первый аргумент командной строки (имя образа программы),
- `task` – номер задачи,
- `res` – возвращаемое значение функции, реализующей решение этой задачи,

Задачи

1. Написать функцию, возвращающую вещественное число, равное машинной точности (т.е. такое максимальное положительное число ε , что $1 + \varepsilon == 1$). Число ε можно искать только в классе чисел вида $1./2^k$.
2. Написать функцию, получающую в качестве аргумента вещественное положительное число x , и возвращающую целое число, равное двоичному порядку x (т.е. такое наибольшее целое число k , что $2^k \leq x < 2^{k+1}$).

3. Написать функцию, получающую в качестве аргумента вещественное число x и возвращающую вещественное число, равное $||x - 1| - 2|/|x - 3|$.
4. Написать подпрограмму, получающую в качестве аргумента три вещественных числа a, b, c , адреса двух вещественных чисел $x1, x2$, и возвращающую по адресу $x1$ максимум из a, b, c , а по адресу $x2$ - минимум из a, b, c .
5. Написать функцию, получающую в качестве аргумента три вещественных числа a, b, c , адреса двух вещественных чисел $x1, x2$, и возвращающую по адресу $x1$ максимальный вещественный корень уравнения $ax^2 + bx + c = 0$, а по адресу $x2$ - минимальный вещественный корень этого уравнения. Функция должна возвращать количество вещественных корней этого уравнения. Если корень один, то он размещается по адресу $x1$, а по адресу $x2$ записывается 0. Если корней 0, то по адресам $x1$ и $x2$ записывается 0. Можно использовать функцию с прототипом

`double sqrt (double);`

для вычисления корня, при компиляции надо добавлять флаг `-lm` в команду компиляции:

`gcc task05.c -lm`

для использования библиотеки математических функций при создании исполнимого файла.

6. Написать функцию, получающую в качестве аргумента целое число n , и возвращающую вещественное число, равное n -му члену последовательности $x_n = 3x_{n-1} + 4 \cdot 2^n, x_0 = 1$.
7. Написать функцию, получающую в качестве аргумента целое число n , и возвращающую вещественное число, равное n -му члену последовательности $x_n = 5x_{n-1} - 6x_{n-2} + 4^{n-1}, x_0 = 2, x_1 = 9$.
8. Написать функцию, получающую в качестве аргумента вещественное число x , целое число n , и возвращающую вещественное число, равное $\sum_{k=1}^n \frac{x^k}{k!}$.
9. Написать функцию, получающую в качестве аргумента вещественное число x , целое число n , и возвращающую вещественное число, равное $\sum_{k=1}^n \frac{(-1)^k x^k}{k}$.
10. Написать функцию, получающую в качестве аргумента целое число n , и возвращающую вещественное число, равное сумме первых n простых чисел.
11. Написать функцию, получающую в качестве аргумента целое число n , и возвращающую целое число, равное n -му простому числу.
12. Написать функцию, получающую в качестве аргумента целое число n , и возвращающую целое число, равное максимальному интервалу между первыми n простыми числами, идущими подряд среди всех простых чисел (интервал между простыми числами p_k и p_{k-1} равен разности $p_k - p_{k-1}$).