

**Ворожцов Артём Викторович**

*Кандидат физико-математических наук,
преподаватель кафедры информатики
Московского физико-технического института (МФТИ),
тренер сборной команды МФТИ по программированию.*

Ситников Владимир Валентинович

*Аспирант, сотрудник кафедры информатики Московского
физико-технического института (МФТИ).*

Тестовые задачи по программированию

Здесь приводятся несколько простых задач по технике программирования. Эти задачи были даны в Московском физико-техническом институте первокурсникам в начале ноября. По сути, контрольная, проведённая по этим задачам, позволила преподавателям проверить умение студентов программировать на языке Си и перейти к более сложным и важным темам, связанным с алгоритмами, структурами данных и проектированием программных систем.

Проверка решений осуществлялась с помощью автоматической системы ejudge. Студенты, используя web-интерфейс, отправляли исходные коды своих программ на языке Си, система автоматически компилировала их и запускала на нескольких входных данных. Задача считалась решённой, если посланная программ-

ма проходила все тесты.

Данные варианты могут рассматриваться школьниками как проверка своей готовности обучаться в МФТИ программированию. Они задают некоторый нижний уровень знаний, который считается обязательным.

Здесь мы приводим 3 варианта (всего в 2007 году было 10 вариантов). В каждом варианте 4 задачи: первые три имеют ценность 1 балл, последняя – два балла. В течение 1,5 часов необходимо набрать 3 балла, т.е. решить три лёгкие или одну лёгкую и одну сложную задачу. Для ввода и вывода данных следует использовать стандартные потоки ввода и вывода. Например, решение задачи «Возведите введённое число в квадрат» выглядит следующим образом:

```

/* автор: Иван Петров
 * задача: Возведение числа в квадрат
 */
#include <stdio.h>
int main() {
    int n;
    scanf ("%d", &n);
    printf ("%d\n", n * n);
    return 0;
}

```

Подсказки и замечания

- Решения должны корректно работать на всех входных данных, удовлетворяющих ограничениям задачи.

- Во многих задачах удобнее использовать тип **unsigned**, который может хранить целые из полуинтервала $[0; 2^{32})$ (это верно для большинства современных компьютеров и компиляторов). Этому типу соответствует формат "%u".

- Следите, чтобы время работы вашей программы не превысило 1 секунду на одном тесте. Можно грубо считать, что 20 млн. простых опера-

ций соответствует одной секунде. Оцените время работы своей программы (желательно экспериментально) на самых больших допустимых входных данных.

- Проверьте работу своей программы как минимум на 5 различных входных данных самостоятельно. Помните, что за неудачную посылку снимается 0,1 балла.

- Не нужно проверять корректность входных данных. Считайте, что входные данные удовлетворяют условиям, описанным в задаче.

Вариант №1

Задача А. (Простые числа) Напишите программу, которая из списка натуральных чисел выделяет простые. Число 1 не считается простым.

Формат ввода. Число N , $N \in \mathbb{N}$,

$0 < N < 10\,000$ и N натуральных чисел из отрезка $[1; 100\,000]$.

Формат вывода. Простые числа из данных в том порядке, в котором они даны на входе.

ВХОД	ВЫХОД
4 3 9 11 3	3 11 3

Задача В. (Число, свободное от квадратов) Напишите программу, которая определяет, есть ли среди делителей числа полные квадраты больше 1.

Формат ввода. Натуральное число N , $0 < N < 2^{31}$.

Формат вывода. Выведите YES (если есть) или NO (иначе).

ВХОД	ВЫХОД
4	YES
12	YES
15	NO

Задача С. (Взвешивание) Напишите программу, которая проверяет, можно ли данное число «взвесить» на двухчашечных весах гирьками с весами $1, 4^1, 4^2, \dots$. Каждая гирька есть только в одном экземпляре. Известно, что числа вида $4 \cdot n + 2$ невозможно взвесить указанным образом. Числа $4 \cdot n, 4 \cdot n + 1, 4 \cdot n - 1$ можно взвесить тогда и только тогда, когда можно взвесить n .

Формат ввода. Натуральное число $N, 0 \leq N < 2^{31}$.

Формат вывода. Одна строка, содержащая YES или NO.



ВХОД	ВЫХОД
59	YES
58	NO

Задача D. (Таблица количеств хороших слов) Напишите программу, которая выводит таблицу чисел $b(n, k)$ ($0 \leq n, k \leq N$), где $b(n, k)$ есть количество бинарных слов длины n , в которых ровно k единиц и нет двух подряд идущих единиц.

Формат ввода. Целое число $N, 0 < N < 48$.

Формат вывода. Выведите $N+1$ строку. Строка с номером n ($n = 0, 1, \dots, N$) содержит числа $b(n, 0), b(n, 1) \dots$ до последнего ненулевого числа включительно.



ВХОД	ВЫХОД
4	1 1 1 1 2 1 3 1 1 4 3

Вариант №2

Задача А. (Транспонирование) Напишите программу, которая транспонирует введённую таблицу целых чисел. Память для хранения чисел выделяйте динамически.

Формат ввода. В первой строке дано натуральное число N – размер таблицы, $0 < N < 50$. В следующих N строках дано по N чисел, разделённых пробелом. Каждое число по мо-

дулю менее 2^{31} .

Формат вывода. Таблица чисел, в которой число, стоящее в i -й строке

ВХОД	ВЫХОД
3	1 4 7
1 2 3	2 5 8
4 5 6	3 6 9
7 8 9	

в j -м столбце, равно числу, стоящему в j -й строке и в i -м столбце данной таблицы.

Задача В. (Сумма делителей)

Напишите программу, которая находит сумму всех делителей натурального числа N , включая 1 и N .

Формат ввода. Натуральное число N , $0 < N < 2^{28}$.

Формат вывода. Сумма делителей числа N , включая 1 и N .

ВХОД	ВЫХОД
5	6
10	18

Задача С. (Нормализация корня)

Нормализацией корня \sqrt{n} , $n \in \mathbb{N}$, назовём представление $\sqrt{n} = b\sqrt{a}$, где $a, b \in \mathbb{N}$ и b максимально. Напишите программу, которая для данного на-

турального числа n находит пару чисел (a, b) .

Формат ввода. Натуральное число n , $1 \leq n < 2^{32}$.

Формат вывода. Выведите числа a и b , разделив их пробелом.

ВХОД	ВЫХОД
24	6 2
36	1 6

Задача D. (Число разложений без повторов)

Напишите программу, которая считает количество разложений $Q(N)$ данного натурального числа N на неупорядоченные натуральные слагаемые без повторов. Например, для $N=5$ есть три различных разложения: $5=5=4+1=3+2$. Разложения считаются различными, если множества слагаемых различаются. Реализуйте функцию $q(n, k)$, возвращающую количество разложений числа n на неповторяю-



щиеся слагаемые, которые меньше либо равны k . Заметьте, что для $1 < k \leq n$ верна формула

$$q(n, k) = q(n - k, k - 1) + q(n, k - 1).$$

Кроме того,

$$q(1, k) = q(0, k) = 1$$

при $k > 0$ и $q(n, k) = 0$ при $k < 0$.

На основе этих соотношений напишите рекурсивную функцию q с

ВХОД	ВЫХОД
5	3
15	27

Вариант №3

Задача А. (НОК трёх чисел) Напишите программу, которая находит наименьшее общее кратное трёх данных чисел.

Формат ввода. Три натуральных

числа a, b и c , $0 < a, b, c < 2^{32}$.

Формат вывода. НОК (a, b, c) . Известно, что НОК (a, b, c) не превосходит $2^{32} - 1$.

ВХОД	ВЫХОД
24 10 3	120
12 45 18	180

Задача В. (Ввод простых чисел по номеру) Напишите программу, которая вычисляет простые числа по их номерам. Число 1 не считается простым.

Формат ввода. Натуральное число N , $0 < N \leq 10\,000$ и N натураль-

ных чисел a_1, \dots, a_N из отрезка $[1; 10\,000]$.

Формат вывода. Выведите N простых чисел p_{a_1}, \dots, p_{a_N} , разделив их пробелом.

ВХОД	ВЫХОД
3	5 3 2
3 2 1	
2	7 919 104 729
1000 10 000	

Задача С. (Сумма четырёх квадратов) Известно, что любое натуральное число N можно представить в виде суммы четырёх квадратов неотрицательных чисел. Напишите программу, которая находит это разложение.

Формат ввода. Натуральное число N , $0 < N < 2\,000\,000$.

Формат вывода. Четыре полных квадрата a^2, b^2, c^2, d^2 , разделённых пробелами, такие, что $a^2 + b^2 + c^2 +$

$$+ d^2 = N, 0 \leq a^2 \leq b^2 \leq c^2 \leq d^2.$$

Если существует несколько раз-
ложений, то выведите одно из них.

ВХОД	ВЫХОД
7	1 1 1 4
107	0 1 25 81

**Задача D. (Максимальный квад-
рат)** Напишите программу, которая
находит на данной квадратной карте
 $N \times N$ белый квадрат максимального
размера. Карта разбита на ячейки
 1×1 . Каждая ячейка покрашена в
чёрный или белый цвет. Реализуйте
рекурсивную функцию

```
int max_square (int x, int y)
с запоминанием вычисленных значе-
ний, которая возвращает размер
максимального белого квадрата с
правой нижней вершиной в ячейке с
координатами (x;y) (координаты уве-
личиваются вниз (y) и вправо (x)).
Подсказка: выразите
max_square(x , y)
через
max_square (x-1, y),
max_square(x , y-1),
max_square (x-1, y-1) и цвет
ячейки с координатами (x;y).
```

Формат ввода. Натуральное чис-
ло $N, 0 < N \leq 100$, а затем N строк по
 N символов. Символ '#' обозначает
чёрный цвет, а символ '.' – белый.

Формат вывода. Размер стороны
максимального белого квадрата и ко-
ординаты его правой нижней верши-
ны. Координаты отсчитываются с ну-
ля. Если есть несколько вариантов
ответа, выведите тот из них, который
имеет меньшую координату x , а при
равных x – тот, который имеет мень-
шую координату y .



ВХОД	ВЫХОД
5 # . . # # # . . . # # . # # #	3 4 3