
· Республика Сербия ·
· Академия технических и художественных прикладных исследований
Белград ·
· Кафедра Высшей школы электротехники и информатики ·
· 2026 ·



ДЕСЯТОЕ ЮБИЛЕЙНОЕ ГОСУДАРСТВЕННОЕ СОРЕВНОВАНИЕ ПО ПРОГРАММИРОВАНИЮ

ТЕКСТЫ ЗАДАЧ

НА РУССКОМ ЯЗЫКЕ

Три задачи · 180 минут · 150 баллов всего

ПРАВИЛА СОРЕВНОВАНИЯ

- Каждая задача стоит максимально **50 баллов**.
- В начале каждой задачи вводятся необходимые входные значения.
- Каждая задача имеет **5 выходов**, вычисляемых на основе введенных входных значений. Каждый выход задачи стоит **10 баллов** если правильный, **0** если неправильный.
- Участники обязаны **ЧЁТКО** обозначить каждый выход меткой подзадачи (напр. **1.1:**, **1.2:** и т.д.). Если выход не обозначен чётко, участник получает 0 баллов за этот выход.
- Если задача остановится из-за ошибки в любой момент, участнику засчитываются выходы, видимые на экране до этого момента. В случае одинакового количества баллов, лучшее место занимает участник, который сдал раньше.
- Соревнование длится **180 минут**.

ЗАДАЧА 1

Ввести два пятизначных числа **a** и **b** (разделённые пробелом, на одной строке). Предполагается, что числа введены правильно – нет необходимости проверять, являются ли введённые числа действительно пятизначными. Каждая подзадача имеет ровно **одну строку вывода** – чётко обозначить её перед выводом (напр. **1.1:...**).

А Суммы, произведения и разность

Вывести в одну строку пять чисел: сумму цифр числа **a**, сумму цифр числа **b**, произведение цифр числа **a**, произведение цифр числа **b**, и абсолютное значение разности суммы цифр числа **a** и суммы цифр числа **b** (в этом порядке, разделённые пробелами).

Б Формирование нового числа **c**

Сформировать новое пятизначное число **c** так, чтобы каждая цифра на позиции i ($i = 0$ для цифры единиц, $i = 1$ для цифры десятков, $i = 2$ для цифры сотен, $i = 3$ для цифры тысяч, $i = 4$ для цифры десятков тысяч) вычислялась по формуле:

$$c[i] = (a[i] + b[i] + i) \bmod 10$$

Вывести только число **c**.

В Массив частот цифр

Сформировать массив **d** из 10 элементов так, чтобы **d[i]** представлял общее количество вхождений цифры i в числах **a**, **b** и **c** вместе. Вывести элементы массива **d** в одну строку, разделённые пробелами (ровно 10 чисел, от **d[0]** до **d[9]**).

Г Матрица **mat** – центральный элемент

Сформировать матрицу **mat** размером 10×10 по следующим правилам:

- Первая строка (строка 0) является массивом **d**.
- Каждая следующая строка i (для $1 \leq i \leq 9$) получается из строки $i-1$ путём циклического сдвига всех элементов предыдущей строки на одну позицию **влево**, затем к каждому элементу прибавляется значение i с сохранением только цифры единиц результата (т.е. $n \bmod 10$).

Вывести только значение центрального элемента: **mat[5][5]**.

Д Вращение матрицы – элемент **rot[2][8]**

Сформировать матрицу **rot**, повернув матрицу **mat** на 90° по часовой стрелке. Вывести только значение элемента **rot[2][8]**.

ЗАДАЧА 2

Ввести строку **s1** длиной **3·N** ($3 \leq N \leq 12$) символов, состоящую из строчных и заглавных букв английского алфавита и цифр. Предполагается, что строка введена правильно. **N** не вводится отдельно – его нужно вычислить из длины строки. Индекс 0 соответствует первому символу строки (напр. **s1[0]** является первым символом). Каждая подзадача имеет ровно **одну строку вывода** – чётко обозначить её перед выводом (напр. **2.1:...**).

А Основная статистика и упорядоченность

Вывести в одну строку четыре значения (разделённые пробелами): количество заглавных букв, количество строчных букв, количество цифр, и слово **DA** если строка **упорядочена**, иначе **NE**. Строка считается упорядоченной, если в ней не существует **двух последовательных символов одного типа** (типы: строчная буква, заглавная буква, цифра).

Б Три последовательности – два символа

Из строки **s1** сформировать 3 последовательности, каждую длиной **N**, так что:

- **sek1** содержит символы с позиций 0, 3, 6, 9, ... (остаток 0 при делении на 3),
- **sek2** содержит символы с позиций 1, 4, 7, 10, ... (остаток 1 при делении на 3),
- **sek3** содержит символы с позиций 2, 5, 8, 11, ... (остаток 2 при делении на 3).

Вывести в одну строку **ровно два символа**, разделённых пробелом: последний символ первой последовательности и первый символ третьей последовательности. Формат: **sek1[N-1] sek3[0]**.

В Преобразование символов – два символа

В каждой из трёх последовательностей выполнить следующее преобразование символов:

- **Строчные буквы** сдвинуть циклически на **3 позиции вперёд** в английском алфавите ('a' → 'd', 'b' → 'e', ..., 'w' → 'z', 'x' → 'a', 'y' → 'b', 'z' → 'c').
- **Заглавные буквы** заменить соответствующими строчными ('A' → 'a', 'B' → 'b', ..., 'Z' → 'z').
- **Цифру** заменить следующей цифрой циклически ('0' → '1', '1' → '2', ..., '8' → '9', '9' → '0').

Преобразование применяется независимо к каждому символу последовательности. Преобразованная последовательность, полученная из **sek1**, обозначается **t1**, из **sek2** – **t2**, из **sek3** – **t3**. Вывести в одну строку **ровно два символа**, разделённых пробелом: **t1[N-1] t3[0]**.

Г Объединение и частоты символов

Объединить преобразованные последовательности (по порядку: первая, вторая, третья) в одну строку **s4**. Вывести в одну строку **четыре значения**, разделённых пробелами: символ, который в **s4** встречается **наибольшее** число раз, это число, символ, который встречается **наименьшее** число раз, и это число. Рассматриваются только символы, встречающиеся в **s4** хотя бы один раз. Если таких символов несколько, взять тот, у которого наименьшее значение ASCII. Формат: **max_kar max_broj min_kar min_broj**.

Д**Квадратная матрица и вращение – элементы `mat[0][K-1]` и `rot[K-1][0]`**

Определить `K = ceil(sqrt(len(s4)))` – размер наименьшей квадратной матрицы `mat`, в которую можно записать `s4` строка за строкой (где `len(s4)` – длина строки `s4`, а `ceil` и `sqrt` – функции округления вверх и квадратного корня). Пустые места заполнить символом `'@'`. Затем сформировать матрицу `rot`, повернув матрицу `mat` на 90° **влево** (против часовой стрелки), по формуле `rot[i][j] = mat[j][K-1-i]`. Вывести в одну строку **три значения**, разделённых пробелами: значение `K`, затем элемент `mat[0][K-1]`, потом элемент `rot[K-1][0]`. Формат: `K mat[0][K-1] rot[K-1][0]`.

ЗАДАЧА 3

Пингвин **Пинго** оказался на ледяной территории, которую контролирует старый морской лев **Лео**. Территория представлена матрицей 10×10 клеток. Начало координат находится в верхнем левом углу матрицы (клетка (0,0)) – **строки и столбцы индексируются от 0 до 9**.

Ввести четыре однозначных числа **r1 c1 r2 c2** (разделённые пробелами, на одной строке). Тем самым заданы две позиции:

- (**r1**, **c1**) – начальная позиция **Пинго** (пингвина),
- (**r2**, **c2**) – начальная позиция **Лео** (морского льва).

Каждая подзадача имеет ровно **одну строку вывода**. Только подзадача **3.5** дополнительно рисует матрицу.

Что такое Манхэттенское расстояние?

Для двух точек (x_1, y_1) и (x_2, y_2) в матричной сетке, **Манхэттенское расстояние** – это сумма абсолютных разностей координат:

$$d = |x_1 - x_2| + |y_1 - y_2|$$

Это **количество шагов**, необходимых для перехода от первой точки ко второй при движении только горизонтально или вертикально (никогда по диагонали) – представьте движение по улицам города с прямоугольными кварталами, отсюда и название (по острову Манхэттен в городе Нью-Йорк).

Пример: между точками (3, 3) и (7, 7) Манхэттенское расстояние равно $|3 - 7| + |3 - 7| = 4 + 4 = 8$ шагов (а не $\sqrt{32} \approx 5,66$, как было бы при евклидовом расстоянии).

Правила игры (упрощённые):

- Пинго движется в направлении **ближайшего края матрицы** (по Манхэттенскому расстоянию от каждой стороны). Если одинаково удалён от нескольких краёв, выбирает направление по приоритету: **ВВЕРХ** (к строке 0), **ВПРАВО** (к столбцу 9), **ВНИЗ** (к строке 9), **ВЛЕВО** (к столбцу 0). Двигается по прямой в этом направлении, по одной клетке за ход. Программа выводит **GORE/DESN0/DOLE/LEVO**.
- Пинго считается **сбежавшим**, если достигает любого края матрицы.
- Лео не двигается в этой упрощённой модели. Исход определяется сравнением расстояний:
 - если **расстояние Пинго до ближайшего края \leq Манхэттенское расстояние между Пинго и Лео**, Пинго сбегаёт,
 - иначе, Лео ловит Пинго на его начальной позиции (**r1**, **c1**).

А

Манхэттенское расстояние

Вывести Манхэттенское расстояние между начальными позициями Пинго и Лео.

Б

Расстояние Пинго от ближайшего края

Вывести наименьшее Манхэттенское расстояние Пинго от любого края матрицы ($\min(r1, 9-r1, c1,$

В

Направление движения Пинго

Вывести слово, обозначающее направление, в котором будет двигаться Пинго, согласно приоритету: **GORE**, **DESN0**, **DOLE** или **LEVO**. Если Пинго уже на краю (расстояние = 0), вывести **IVICA**.

Г**Исход и конечная позиция**

Вывести исход и конечную позицию Пинго на одной строке. Формат:

- если сбежал: **PINGVIN POBEGAO NA (X,Y)**,
- если пойман: **LAV UNVATIO NA (X,Y)**.

Д**Матрица ответа с метками**

Вывести матрицу **10×10** с метками строк и столбцов (числа 0–9).

Использовать символы:

- **'P'** – начальная позиция Пинго (клетка (**r1**, **c1**)),
- **'L'** – позиция Лео (клетка (**r2**, **c2**)),
- **'p'** – путь, по которому прошёл Пинго (без начальной и конечной позиций),
- **'F'** – конечная клетка, если Пинго сбежал (на краю),
- **'X'** – клетка, где Лео поймал Пинго (совпадает с **'P'** – в этом случае пишется только **'X'**),
- **'-'** – все остальные клетки.

Формат вывода: перед каждой строкой написать её номер, а над матрицей – строку с номерами столбцов (см. тестовый пример).

★ Удачи на соревновании! ★