

## День 2

## Общее описание задач

№	Название задачи	Ограничение на объем используемой памяти	Ограничение на время выполнения программы, секунды	Количество баллов присвоенных задаче
1.	Покупки	$\leq 1\text{Mb}$	$\leq 15,0$	100
2.	Кратное трём	$\leq 1\text{ Mb}$	$\leq 0,1$	100
3.	Компьютерные шахматы	$\leq 1\text{Mb}$	$\leq 0,1$	100

*Примечание.* При равенстве общего числа очков, лучшим будет считаться решения с меньшим временем выполнения. В случае равенства и времени выполнения, лучшим будет считаться решение, использующее меньший объем памяти.

## Покупки

Ион отправился за покупками в торговый центр. Торговый центр представляет собой многоэтажное здание в форме параллелепипеда, состоящего из помещений размером  $1 \times 1 \times 1$  единиц длины. Параллелепипед имеет размеры  $N \times L \times C$ , где  $N$  – число этажей здания,  $L$  – длина, а  $C$  – ширина здания.

Клиентам доступны не все помещения, так как часть из них используются для хранения товаров и проход покупателей через такие помещения запрещен.

Покупатели могут проходить из одного доступного помещения в соседнее доступное помещение двигаясь вперед, назад, влево или вправо, поднимаясь или опускаясь на один этаж по эскалатору. Такие эскалаторы находятся в каждом из свободных для прохода покупателей помещений торгового центра.

Проход из одного свободного помещения в другое соседнее свободное помещение, подъем либо спуск на один этаж занимает ровно одну минуту.

Рассматривая необходимые ему товары, Ион вдруг вспомнил что опаздывает на деловую встречу. Следовательно, ему нужно как можно быстрее выйти из торгового центра. Очевидно, для этого, ему следует дойти как можно быстрее до любого из свободных помещений в котором есть выход из здания торгового центра.

**Задача.** Напишите программу, которая возвращает минимальное число минут, необходимых Иону для перехода из помещения в котором он находится, в любом из помещений с выходом из торгового центра, а также число всевозможных путей с таким временем перехода.

**Входные данные.** Стандартный ввод содержит в первой строке целые числа  $N$ ,  $L$  и  $C$ , разделенные пробелами. Далее, на  $L$  строках следует план первого этажа. Каждая из таких строк содержит по  $C$  символов, описывающих помещения рассматриваемого этажа. Далее, на следующих  $L$  строках стандартного ввода описан план второго этажа и т.д.

Помещения через которые возможен проход указаны цифрой 0, помещения, через которые нельзя проходить указаны буквой M. Помещение, в котором находится Ион, указано буквой D, а помещения в котором есть выходы из здания указаны буквами I.

**Выходные данные.** Стандартный вывод должен содержать одну строку с двумя целыми числами, разделенными пробелом: минимальное число минут необходимых для перехода Иона из помещения в котором он находится в любом из помещений с выходом из торгового центра, а также число всевозможных путей с таким временем перехода.

**Ограничения.**  $1 < N, L, C \leq 50$ . Гарантируется существование хотя бы одного выходного пути. Ограничения на время выполнения программы и на объем используемой памяти приведены в общем описании предложенных для решения задач. Исходный файл должен иметь имя `cumparaturi.pas`, `cumparaturi.c` или `cumparaturi.cpp`.

### Пример.

*Ввод*

```
3 4 5
0000D
0MMM0
0MM00
MMMM0M
MMMMM
MMMMM
MM0MM
MM000
MMMMM
MMMMM
M0MMM
MM00I
```

*Вывод*

```
7 2
```

### Кратное трём

Рассматривается натуральное число  $N$ , записанное без незначимых нулей. Обозначим через  $M$  натуральное число, получаемое из исходного числа  $N$  путем замены одной из цифр, находящейся на любой позиции рассматриваемого числа, на некоторую другую цифру. Поскольку существует несколько способов таких замен, возможно, что среди полученных чисел  $M$  есть и такие, которые делятся без остатка на 3 (Кратное трём). Требуется вычислить наибольшее из полученных чисел, кратные трём, которое мы обозначим через  $M_{max}$ .

Например, допустим  $N = 123$ .

Заменяем цифру "1" на цифру "9". Полученное число  $M = 923$  не делится без остатка на 3.

Заменяем цифру "1" на цифру "8". Полученное число  $M = 823$  не делится без остатка на 3.

Заменяем цифру "1" на цифру "7". Полученное число  $M = 723$  делится без остатка на 3.

Очевидно, что любые следующие из замен цифры "1" на цифры "6", "5", ..., "2" дадут числа меньше чем 723. Еще меньшими будут числа  $M$ , получаемые путем замены каждой из оставшихся цифр "2" и "3" числа  $N$ .

Следовательно,  $M_{max} = 723$ .

**Задание.** Напишите программу, которая, зная число  $N$ , находит число  $M_{max}$ .

**Входные данные.** Первая строка стандартного ввода содержит целое число  $N$ .

**Выходные данные.** В единственной строке стандартного вывода должно содержаться целое число  $M_{max}$ .

**Ограничения.**  $1 \leq N \leq 10^{18}$ . Ограничения на время выполнения программы и на объем используемой памяти приведены в общем описании предложенных для решения задач. Исходный файл должен иметь имя `multiplu.pas`, `multiplu.c` или `multiplu.cpp`.

#### Пример 1.

*Ввод*

123

*Вывод*

723

#### Пример 2.

*Ввод*

7878

*Вывод*

7875

## Компьютерные шахматы

Известно, что в случае классических шахмат игра происходит на доске квадратной формы, разделенной на 8 горизонтальных рядов (горизонтали) и 8 вертикальных столбцов (вертикали), которые образуют 64 квадрата с равными поверхностями, называемыми полями. Поля окрашены попеременно в черный и белый цвета (см. *Рисунок 1*).

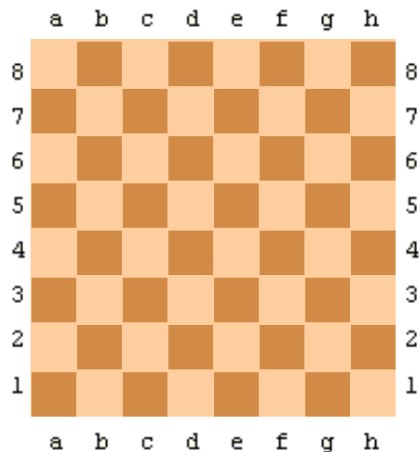


Fig. 1

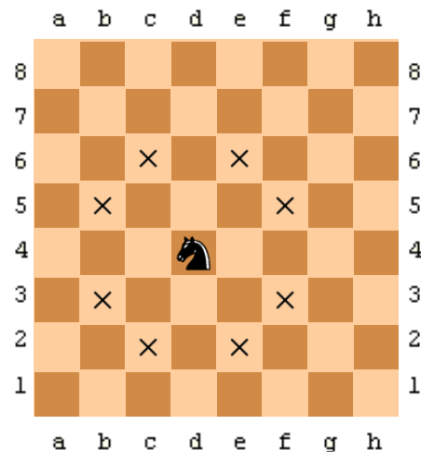


Fig. 2

Горизонтали классической шахматной доски пронумерованы  $1, 2, 3, \dots, 8$ , а вертикали – строчными буквами латинского алфавита  $a, b, c, \dots, h$ . Поля обозначаются буквой вертикали, за которой следует номер горизонтали. Например, поле, на котором расположен конь показанный на *Рисунке 2*, обозначается как  $d4$ .

В случае *компьютерных шахмат* доска также имеет квадратную форму, но может содержать до 26 вертикалей, обозначенных буквами  $a, b, c, \dots, z$ , и, соответственно, до 26 горизонталей, обозначенных номерами  $1, 2, 3, \dots, 26$ . Поля указываются с помощью символьных строк вида:

$\langle \text{Вертикаль} \rangle \langle \text{Горизонталь} \rangle$

где  $\langle \text{Вертикаль} \rangle$  – одна из строчных букв латинского алфавита, а  $\langle \text{Горизонталь} \rangle$  – один из номеров  $1, 2, 3, \dots, 26$ .

Примеры:  $d4, d21, y5, y12$ .

Каждый из игроков в компьютерные шахматисты имеет ровно по одной фигуре: первый игрок имеет одного белого коня, а второй игрок – одного черного коня.

Как и в случае классических шахмат, ходить конем в случае компьютерных шахмат можно только в форме буквы «L», как показано на *Рисунке 2*. На этом рисунке поля, на которые за один ход может перейти конь с поля  $d4$ , помечены знаком «x».

В отличие от классических шахмат, в которые игроки ходят последовательно, друг за другом, и на каждом поле может находиться ровно одна фигура, в компьютерных шахматах оба игрока ходят одновременно, а на одном поле могут находиться не только одна, но и две фигуры.

В случае классических шахмат, ход фигурой описывается символьной строки вида:

$\langle \text{Ход фигурой} \rangle ::= \langle \text{Вертикаль } 1 \rangle \langle \text{Горизонталь } 1 \rangle - \langle \text{Вертикаль } 2 \rangle \langle \text{Горизонталь } 2 \rangle$

где:

$\langle \text{Вертикаль } 1 \rangle \langle \text{Горизонталь } 1 \rangle$  – указывает поле, с которого снимается шахматная фигура;

$\langle \text{Вертикаль } 2 \rangle \langle \text{Горизонталь } 2 \rangle$  – указывает поле, на котором ставится шахматная фигура.

Например, в случае *Рисунка 2*, ход конем с поля  $d4$  на поле  $f5$  описывается с помощью символьной строки  $d4-f5$ , а ход конем с того же поля на поле  $c2$  описывается с помощью символьной строки  $d4-c2$ .

В случае компьютерных шахмат, совместный ход коней описывается символьной строки вида:

$\langle \text{Ход конями в компьютерных шахматах} \rangle ::= \langle \text{Ход белым конем} \rangle \& \langle \text{Ход черным конем} \rangle$

Например, обозначение:

$d4-f3 \& a1-b3$

имеет следующий смысл: белый конь переходит с поля  $d4$  на поле  $f3$ , а черный конь – с поля  $a1$  на поле  $b3$ .

Будучи компьютерным шахматистом, Ион решил разработать стратегию, которая обеспечила бы ему как можно больше выигрышей. Для начала, ему нужна программа, которая вычисляет минимальное число ходов, необходимых для того, чтобы оба коня – и черный, и белый – оказались вместе на одном и том же поле, неважно на каком. Подчеркиваем, что в при каждом ходе игры в компьютерные шахматы, оба коня – и белый, и черный – обязательно должны изменить свои позиции на шахматной доске..

**Задание.** Напишите программу, которое, зная исходные поля на которых находятся кони, вычисляет минимальное количество ходов, необходимых для того, чтобы оба коня оказались вместе на одном из полей, неважно на каком.

**Входные данные.** Стандартный вход содержит в первой строке целое число  $N$  – число горизонталей квадратной компьютерной шахматной доски. Вторая строка стандартного ввода содержит две символьные строки, разделенные пробелом. Первая строка указывает поле на котором находится белый конь, а вторая – поле, на котором находится черный конь.

**Выходные данные.** Стандартный вывод должен содержать в одной строке целое число  $M$  – минимальное количество ходов, необходимое для перемещения обоих коней на одно и то же поле, неважно на какое. Если это невозможно, в стандартный вывод следует записать целое число «-1».

**Ограничения.**  $1 < N \leq 26$ . Ограничения на время выполнения программы и на объем используемой памяти приведены в общем описании предложенных для решения задач. Исходный файл должен иметь имя `sahul.pas`, `sahul.c` или `sahul.cpp`.

### Пример 1.

*Ввод*

a1 a3

*Вывод*

1

*Пояснение*

Достаточен ровно один ход:  
 $a1-c2 \& a3-c2$ .

### Пример 2.

*Ввод*

a1 a1

*Вывод*

-1

*Пояснение*

Не существует ни одной последовательности ходов, которая удовлетворяла бы условиям задачи.