1. Максимальная цифра

Дано целое неотрицательное число, состоящее не более чем из 5 цифр. Найти значение самой большой цифры в числе и их количество.

Входные данные

В первой и единственной строке входных данных записано целое число n ($0 \le n \le 99999$).

Выходные данные

В единственной строке вывести значение максимальной цифры и количество таких цифр в числе.

Примеры

входные данные

```
24341
```

выходные данные

```
4 2
```

Примечание

Самая большая цифра числа равна 4, и их две.

Идея решения

Задачу можно решать через введение данных как строки и поиск количества максимальных по коду символов в строке. Пусть мы не знаем строк. Тогда выделить какую-нибудь цифру числа (а именно последнюю) можно через остаток от деления числа на 10. Цифру необходимо сравнить с максимальной. Если она равна максимальной, то количеств максимальных цифр увеличить на 1. Если она больше максимальной, то запомнить новую максимальную цифру и сделать её количество равной 1. После обработки последней цифры её нужно убрать. Это можно получить, разделив число нацело на 10. Таким образом можно перебрать все цифры числа (циклом, если мы их знаем, или вручную, скопировав нашу последовательность действий ещё 4 раза). Отдельно нужно рассмотреть случай, если наше число равно 0, чтобы 0 не посчитался больше 1 раза.

Пример программы решения без строк и циклов:

```
Program z1;
varx, a, max, nmax : longint;
begin
  read(x);
a := x mod 10; x := x div 10;
  max := a; nmax := 1;
a := x mod 10; x := x div 10;
if a > max then begin max := a; nmax := 1 end else if a = max then nmax := nmax + 1;
a := x mod 10; x := x div 10;
if a > max then begin max := a; nmax := 1; end else if a = max then nmax := nmax + 1;
a := x mod 10; x := x div 10;
if a > max then begin max := a; nmax := 1; end else if a = max then nmax := nmax + 1;
a := x mod 10; x := x div 10;
if a > max then begin max := a; nmax := 1; end else if a = max then nmax := nmax + 1;
if max = 0 then nmax := 1;
write(max, ' ', nmax);
end.
```

2. Автобусы

Из лагеря отдыха «Бригантина» возвращаются 4отряда ребят (в каждом отряде не менее 10 и не более 25 человек). В лагере они занимались исследовательской работой, и к концу смены разделились на математиков, физиков и информатиков. Для того, чтобы в автобусе было интереснее ехать, ребята хотят сесть в автобусы так, чтобы там были или только математики, или только физики. Кроме того, ни математики, ни физики не против того, чтобы с ними ехали информатики. Какое минимальное количество автобусов понадобится для перевозки ребят так, как они хотят, если в автобус помещается не более *n*ребят?

Входные данные

В первых 4 строках содержатся по 3 неотрицательных целых числа a_i,b_i, c_i, содержащие информацию о количестве соответственно математиков, физиков и информатиков в очередном отряде. В девятой строке записано число $n(1 \le n \le 40)$— количество посадочных мест в автобусе.

Выходные данные

В единственной строке выведите одно число — минимальное количество автобусов, необходимое для перевозки ребят с разделением на математиков и физиков.

Примеры

входные данные

```
8 8 8
0 10 10
10 0 12
5 15 5
19
```

выходные данные

```
5
```

Примечание

Одна из возможных рассадок: (18,0,1)(0, 18,1) (5, 0, 14) (0,15,4) (0, 0, 15).

Идея решения

Для начала необходимо вычислить, сколько у нас всего учеников по каждому из предметов. Для этого после ввода информации по каждому из отрядов прибавляем количество математиков этого отряда к суммарному количеству математиков всего, так же с физиками и информатиками (можно делать в цикле, если не знаем циклов, то копируем набор операторов 3 раза). Зная общее количество математиков, посчитаем, сколько автобусов понадобится для того, чтобы полностью наполнить их математиками (целая часть от деления количества математиков на количество мест в автобусе). Точно так же поступим с физиками. Если у нас остались неразмещённые математики (остаток от деления математиков на количество мест больше 0), то выделяем для них ещё автобус, дополняя пустые места информатиками. Аналогично для физиков. Оставшихся информатиков сажаем в автобусы (количество информатиков делим на количество мест, округляем вверх).

Пример программы решения с циклами:

```
Program z1;
var
a, b, c, x, y, z, n, i, k : longint;
begin
a := 0; b := 0; c := 0;
fori := 1 to 4 do
begin
read(x, y, z);
a := a + x; b := b + y; c := c + z;
end;
read(n);
k := a div n + b div n; a := a mod n; b := b mod n;
if a > 0then begink := k + 1; c := c - (n - a);end;
if b > 0 then begin k := k + 1; c := c - (n - b); end;
if c > 0 then begin
k := k + c div n;
if c mod b > 0 then k := k + 1;
end;
write(k);
end.
```

3. Альфаралли

На планете Альфа все города расположена на экваторе, расстояние между всеми соседними городами одинаковое, равное 1. Города названы прописными буквами латинского алфавита, по порядку, от A до Z. Единственная дорога планеты проходит по экватору, соединяя города между собой. Так что соседними городами являются те города, буквы латинского алфавита которого находятся рядом (города A и Zтакже являются соседними).

Жители планеты решили организовать ралли. В задачу участнику необходимо отмечаться в городах планеты в определённом порядке. При этом города, не входящие в порядок, просто пропускаются (в них нет необходимости останавливаться, но тем не менее проехать через них надо).

Определить минимальное расстояние, которое необходимо преодолеть участнику.

Входные данные

В единственной строке ввода задана непустая строка из прописных символов латинского алфавита.Длина строки не превышает 256 символов.

Выходные данные

Единственная строка выходного файла должна содержать минимальное расстояние, которое необходимо преодолеть участнику ралли.

Примеры

входные данные

```
RALLY
```

выходные данные

```
33
```

входные данные

```
ZAXAN
```

выходные данные

```
20
```

Примечание

В первом примере, например, два раза подряд нужно отметиться в городе L–значит, из него можно не уезжать после первой отметки, а сразу поставить вторую. Суммарное расстояние: RA (9) + AL (11) + LL (0) + LY(13) = 33.

Во втором примере: ZA (1) + AX (3) + XA(3) + AN (13) = 20

Идея решения

Для каждой пары символов, представляющей собой часть маршрута, необходимо посчитать кратчайший путь – пересекая границу ZA или нет. Длина пути без пересечения границы равна модулю разности кодов символов, с пересечением границы = 26 – модуль разности. Берём меньшее из значений, добавляем к сумме. Повторяем для всех пар рядом стоящих символов.

Пример программы решения с циклами:

```
Program z1;
var
s : string;
i, k, t : longint;
begin
read(s);
k := 0;
fori := 1 to Length(s)-1 do
begin
t := abs(ord(s[i]) - ord(s[i+1]));
if t > 26 - t then t := 26 - t;
k := k + t;
end;
write(k);
end.
```

4. Карандаши

У Андрея есть большая коробка, в которой лежит Nцветных карандашей. Он собирается сделать подарки своим одноклассникам. Для этого он хочет каждому из одноклассников подарить набор из Мкарандашей. Чтобы набор оказался красивым, нужно, чтобы все карандаши в нём были разных цветов. Какое максимальное количество красивых наборов может собрать Андрей?

Входные данные

В первой строке входных данных записано два целых числа: nиm ($1 \le n \le 1000$, $1 \le m \le 1000$). Во второй строке перечислены nцелых чисел (в пределах от 1 до 1000) - номера цветов карандашей в коробке.

Выходные данные

Единственная строка, содержащая одно число — максимальное количество красивых наборов, которое можно собрать.

Примеры

входные данные

```
7 3
1 4 1 2 2 3 5
```

выходные данные

```
2
```

входные данные

```
10 2
1 1 1 1 1 1 1 1 1 1
```

выходные данные

```
0
```

Примечание

В первом примереможно собрать наборы 1 4 2 и 1 2 3. Во втором нет разных по цвету карандашей.

Идея решения

Первое, что приходит в голову – «жадный» алгоритм. То есть на каждом шаге алгоритма мы должны взять в набор те карандаши, каких у нас больше всего. Для того, чтобы каждый раз выбирать Ммаксимумов, необходима сортировка. Если сортировка не очень эффективна (например, «пузырьковая»), то времени может не хватить. Поэтому нужно использовать либо эффективную сортировку (быстрая, подсчётом), либо использовать подходящую структуру данных (приоритетная очередь, бинарная куча).