

1.

Выпишем все буквы русского алфавита без «Ё» и «Й», но с добавлением пробела «\_». Каждой букве соответствует десятичный порядковый номер – её код.

	А	Б	В	Г	Д	Е	Ж	З	И	К	Л	М	Н	О	П
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Приведём лишь часть таблицы, где буквам соответствуют двоичные пятизначные коды:

А	Б	В	Г	...	Э	Ю	Я
00001	00010	00011	00100	...	11101	11110	11111

Главная проблема столь простого способа передачи информации заключается в защите передаваемых данных. Один из способов защиты информации – это ввод ключа, или кодового слова, который складывают с кодом сообщения перед его посылкой по каналу связи. Сложение производится в двоичной системе счисления по правилам:  $0 + 0 = 0$ ,  $1 + 0 = 1$ ,  $1 + 0 = 1$ ,  $1 + 1 = 0$  (такая операция называется сложением по модулю два: обратите внимание, что при сложении единиц в старший разряд ничего не добавляется).

Приведём пример. Пусть дана строка: **БУДУ ДОМА**, кодовое слово: **КЛЮЧ**

Б	У	Д	У	_	Д	О	М	А
00010	10011	00101	10011	00000	00101	01110	01100	00001
01010	01011	11110	10111	01010	01011	11110	10111	01010
К	Л	Ю	Ч	К	Л	Ю	Ч	К
01000	11000	11011	00100	01010	01110	10000	11011	01011
З	Ш	Ь	Г	К	О	Р	Ы	Л

Результат кодирования: **ЗЩЬГКОРЫЛ**

Без кодового слова восстановить исходное сообщение невозможно. С другой стороны, алгоритм расшифровки очень прост – достаточно снова прибавить к сообщению ключ по тем же правилам:

З	Ш	Ь	Г	К	О	Р	Ы	Л
01000	11000	11011	00100	01010	01110	10000	11011	01011
01010	01011	11110	10111	01010	01011	11110	10111	01010
К	Л	Ю	Ч	К	Л	Ю	Ч	К
00010	10011	00101	10011	00000	00101	01110	01100	00001
Б	У	Д	У	_	Д	О	М	А

Вам дан файл с четным количеством строк, каждая строка не длиннее, чем 250 символов: в каждой паре строк первая строка – сообщение, требующее кодирования, а вторая – ключ. Сформировать текстовый файл из закодированных строк.

Ввод	Вывод
БУДУ ДОМА КЛЮЧ БУБСМЦГААРЛ ЦЦ АБРАКАДАБРА	ЗЩЬГКОРЫЛ ВСТРЕЧА В КАФЕ

2.

Упаковка информации методом RLE-кодирования состоит в следующем. Упакованная последовательность содержит управляющие байты, за каждым управляющим байтом следует один или несколько байтов данных. Если старший бит управляющего байта равен 1, то следующий за управляющим байт данных нужно повторить столько раз, сколько записано в оставшихся 7 битах управляющего байта. Если же старший бит

управляющего байта равен 0, то надо взять несколько следующих байтов данных без изменения. Сколько именно – записано в оставшихся 7 битах управляющего байта. Например, управляющий байт 10000111 говорит о том, что следующий за ним байт надо повторить 7 раз, а управляющий байт 00000100 – о том, что следующие за ним 4 байта надо взять без изменений.

По каналу связи получена последовательность байт, если интерпретировать их как однобайтовые беззнаковые целые числа, то она записывается в виде последовательности неотрицательных чисел, записанных в текстовом файле. Первое число в файле – это количество байт (чисел) в полученной последовательности.

Известно, что эта последовательность перед передачей была закодирована RLE методом. Раскодируйте последовательность, определите и запишите в ответ количество байт в полученной последовательности и сами числа.

Ввод	Вывод
8	12
133 55 3 147 144 128 132 33	55 55 55 55 55 147 144 128 33 33 33 33

3. При подготовке космического корабля к полёту требуется перевести из института в ангар большое количество грузов, в том числе и дорогостоящие приборы. Для этих целей выделен контейнер размером  $L \times W \times H$ , где  $L$ ,  $W$  и  $H$  – длина, ширина и высота контейнера в метрах (величины целочисленные). Имеется  $N$  одинаковых коробок размером  $X \times Y \times Z$ , где  $Z$  – это высота коробки, а  $X$  и  $Y$  – размеры её прямоугольного дна в метрах (все значения целочисленные). Все коробки маркированы знаком, запрещающим переворачивать коробки кверху дном или ставить коробки на бок. Разрешено ставить коробки друг на друга. Напишите программу, которая будет вычислять минимальное количество «этажей» коробок в контейнере. Если все коробки не поместятся в один контейнер, то вывести 0.

Порядок ввода:

в первой строке размеры контейнера  $L W H$ ;

во второй строке количество коробок  $N$ ;

в третьей строке размеры коробок  $X Y Z$ .

Вход	Выход
10 8 4	2
11	
3 4 1	

4. Луноход перемещается по прямой линии, вдоль которой от точки старта проложена ось  $OX$ . Луноход пересекает кратеры, имеющие форму кругов. Для каждого кратера луноход фиксирует  $x$ -координату центра кратера  $X_i$  и длину хорды  $D_i$  – отрезок линии пересечения трассы лунохода и кратера. Величина  $D_i$  положительна, т.е. луноход не распознаёт кратеры, граница которых только касается трассы. Сопоставим каждой точке трассы её «кратность» – количество кратеров, которым принадлежит эта точка. Например, если точка касания двух кратеров лежит на трассе лунохода (и другим кратерам эта точка не принадлежит), то «кратность» этой точки равна двум. Накопленная информация передается на Землю в следующем формате:  $N$  – количество кратеров, определённых за время движения ( $0 < N \leq 100$ ), а далее  $N$  строк, в каждой строке два вещественных числа  $X_i$  и  $D_i$ , разделённые пробелом. Написать программу, которая по полученным от лунохода данным вычисляет максимальную «кратность» точек трассы лунохода.

Пример.

Вход	Выход
6	3
1 1	
9 2	
3.0 4	
3.5 1	
6 2	
3 1	
3	1
20 1	
2 6	
8 2	

5. Помогите программисту, который работает над очередной 3-хмерной игрой. Имеется подвал со сложной конфигурацией стен. Если посмотреть на план подвала (вид сверху), то стены представляют собой замкнутую ломаную без самопересечений. Эта ломаная задается  $N$  парами координат своих вершин в порядке обхода ограничиваемой забором области против часовой стрелки. Вершины пронумерованы от 1 до  $N$ ,  $N < 100$ .

В точке  $(x, y)$  стоит наблюдатель ( $(x, y)$  не может лежать на ломаной). Считая, что каждому звену ломаной становится в соответствие пара номеров концевых вершин, написать программу, которая указывает, какие звенья наблюдатель увидит полностью или частично в качестве невырожденного отрезка, а какие - вообще нет. Если при взгляде звено видно как точка или как пара, точек, то полагаем, что оно не видно.

Итак, входные параметры, на первой строке - два числа - координаты наблюдателя, на второй строке  $N$  - количество вершин ломаной. Затем  $N$  строк, содержащих пары чисел - координаты вершин. Выходные данные - строки, содержащие по два числа - номера вершин, образующих видимые полностью или частично стены подвала.

Например,

Вход	Выход
5 8	1 2
8	2 3
3 3	4 5
3 8	5 6
5 11	6 7
5 14	1 8
7 14	
7 7	
5 5	
7 3	

6.

Карту местности можно представить в виде целочисленной матрицы  $D(N, M)$  (размерность матрицы не превосходит  $100 \times 100$ ). Каждый элемент матрицы описывает состояние соответствующего квадрата местности и может принимать значения: 0 - поляна или время в часах, необходимое для расчистки участка, от 1 до 5.

Группа туристов высаживается в квадрате с координатами  $(1, 1)$  и должна пройти в квадрат с координатами  $(N, M)$ . Переход от одного квадрата к другому может происходить только через вертикальную или горизонтальную границу между квадратами, группа может двигаться только влево или вниз. Длина пути группы измеряется количеством пройденных квадратов, включая начальный и конечный квадрат.

Требуется составить программу, которая находит для данной карты местности такой путь группы, при котором на расчистку тратится наименьшее время. Если таких путей несколько, то можно вывести любой.

Вывести также время, которое необходимо для прохождения найденного пути.

**Порядок вывода результатов:**

**Порядок ввода исходных данных**

N M

D(1, 1) D(1, 2) ... D(1, M)

D(2, 1) D(2, 2) ... D(2, M)

...

D(N, 1) D(N, 2) ... D(N, M)

Время\_расчистки

LEFT Y1

X2 BOTTOM

...

Xk Yk

**Например:**

<b>Вход</b>	<b>Выход</b>
3 5	10
1 2 1 0 4	1 1
4 1 0 2 2	1 2
1 2 3 3 2	2 2
	2 3
	2 4
	2 5
	3 5