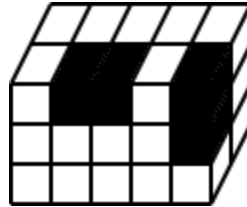


1. КУБИКИ

10 очков

3 секунд

Из прозрачных и черных кубиков собран прямоугольный параллелепипед, размеры которого $N \times M \times K$ кубиков. Даны три проекции этого параллелепипеда – вид спереди, вид сверху и вид справа. В каждой проекции прозрачными являются те места, в которых целый столбец (по всей толщине параллелепипеда) состоит из прозрачных кубиков. Написать программу, которая определяет наибольшее возможное количество черных кубиков в этом параллелепипеде.



Входные данные. На первой строке текстового файла KUUBID.SIS даны размеры параллелепипеда: высота N , ширина M и толщина K ($1 \leq N, M, K \leq 30$). На каждой из следующих N строк записано по M символов – вид спереди, где 0 означает прозрачное, а 1 – непрозрачное место. Вид спереди дан по слоям параллелепипеда сверху вниз, каждый слой слева направо. Далее аналогично дан вид сверху: на K строках записано по M символов. Вид сверху дан по слоям начиная от заднего к переднему, слева направо. Наконец, на последних N строках дан вид справа – по K символов на строке, по слоям сверху вниз, каждый слой спереди назад. (Пример входных данных соответствует изображённому на рисунке параллелепипеду.)

Выходные данные. В текстовый файл KUUBID.VAL вывести одно число – наибольшее возможное число черных кубиков в параллелепипеде.

Пример.

KUUBID.SIS	KUUBID.VAL
3 5 2	4
01101	
00001	
00000	
00000	
01101	
10	
10	
00	

2. ТЕКСТОВЫЙ РЕДАКТОР

20 очков

3 секунд

Реализовать программируемый текстовый редактор, который способен совершать над данным ему текстом следующие операции:

COPY X	скопировать из текста строку X ($1 \leq X \leq N$) в свой буфер обмена
CUT X	удалить из текста строку X ($1 \leq X \leq N$) и поместить её в буфер
PASTE X	вставить перед строкой X ($1 \leq X \leq N+1$) строку из буфера
SPLIT X Y	разделить строку X ($1 \leq X \leq N$) надвое перед символом Y ($1 \leq Y \leq L$)
JOIN X	соединить строки X ($1 \leq X \leq N-1$) и $X+1$ в одну строку (N – число строк в тексте, L – длина строки X)

Входные данные. На первой строке текстового файла TEKST.SIS записано число N ($1 \leq N \leq 100$) строк текста, подлежащего обработке. На следующих N строках записаны строки текста, состоящие из больших и маленьких латинских букв и пробелов. Длина каждой строки не превышает 50 символов. На следующей строке записано количество операций K ($1 \leq K \leq 100$) и далее на K строках команды перечисленных типов – по одной на строке.

Выходные данные. На первую строку текстового файла TEKST.VAL вывести число строк M полученного в результате текста и на следующие M строк вывести этот текст. Известно, что в процессе выполнения команд длина ни одной строки не превысит 100 символов и что

изначально в буфере обмена находится пустая строка (т.е. строка, не содержащая символов; после выполнения операции PASTE в таком случае в текст в указанном месте вставляется эта пустая строка – количество строк после этого увеличится на 1).

<u>Пример.</u>	TEKST.SIS	TEKST.VAL
	2	3
	aaabbb	bbbccc
	ccc	aaa
	4	bbbccc
	SPLIT 1 4	
	JOIN 2	
	COPY 2	
	PASTE 1	

3. ЛИНКЕР

30 очков

3 секунд

Линкер – это настольная игра для одного игрока. Игровой стол состоит из полей, соединённых между собой стрелками. К каждой стрелке приписано целое число. Стрелки размещены таким образом, что начав с любого поля и двигаясь по стрелкам невозможно прийти обратно на это поле.

Игрок помещает свою фишку на поле А. Двигаясь по стрелкам, ему необходимо добраться до поля В. Число очков игрока – это произведение всех чисел на стрелках, по которым прошёл игрок. Написать программу, которая находит максимально достижимое число очков и соответствующую последовательность ходов.

Входные данные. На первой строке текстового файла LINKER.SIS дано количество N ($2 \leq N \leq 100$) полей на столе и число M ($1 \leq M \leq 1000$) соединяющих их стрелок. Поля пронумерованы числами от 1 до N . На следующих M строках записаны данные о стрелках в виде $V_1 V_2 X$ где V_1 – начальное, V_2 – конечное поле этой стрелки, а X – целое число, по абсолютному значению не превышающее 100, приписанное к этой стрелке. На последней строке файла даны числа A и B – поле, с которого игрок начинает ходить, и поле, на котором он должен закончить.

Выходные данные. На первую строку текстового файла LINKER.VAL вывести наибольшее достижимое число очков, на вторую строку – число K полей, которое игрок пройдёт по пути (включая начальное и конечное поля), а на следующие K строк – номера полей в порядке их прохождения игроком. Входные данные всегда такие, что найдётся путь из поля A в поле B . Если возможных решений несколько, вывести любое из них.

<u>Пример.</u>	LINKER.SIS	LINKER.VAL
	4 5	6
	1 3 3	3
	1 4 4	1
	1 2 2	2
	4 3 1	3
	2 3 3	
	1 3	

4. CLICKOMANIA

40 очков

Clickomania – это компьютерная игра для одного игрока. Игровое поле изначально состоит из $N \times M$ разноцветных квадратов. Если два квадрата одного цвета имеют общую сторону, то они принадлежат одному кластеру (причём единичный квадрат, не имеющий соседей того же цвета по горизонтали или вертикали кластера не образует). Если игрок щёлкает мышью по любому квадрату из какого-либо кластера, то все квадраты, принадлежащие этому кластеру убираются с поля, и на их места падают те квадраты, которые до этого лежали на них (так, что ни под одним квадратом не остаётся пустого места). Если какой-либо столбец становится пустым, то столбцы справа от него сдвигаются влево чтобы заполнить пустое место.

12313267	12313267	12 7	1 7	1 7
23555767	23 767	23 1 267	22 267	22 267
24545556	24 4 6	24 4 766	23 1 766	231766
23535676	23 3 676	23333676	24 4 676	244676
Рис. А	Рис. В	Рис. С	Рис. D	Рис. Е

Например, в приведённом на рисунке А положении все квадраты, цвет которых обозначен цифрой 5, образуют один кластер, и если игрок щёлкнет на любом из них, то все они пропадут с игрового поля (рис. В), после чего на их место упадут квадраты сверху (рис. С). В получившемся положении кластер образуют, например, квадраты в нижнем ряду, обозначенные цифрой 3. После того, как игрок щелкнет на них, они пропадут (рис. D) и все столбцы, кроме двух первых сдвинутся влево чтобы заполнить место. Теперь получится положение показанное на рис. Е и игрок может опять щелкать на выбранных квадратах.

За убирание каждого кластера игрок получает $(R-2)^2$ очков, где R – число принадлежащих этому кластеру квадратов. Кроме того игроку начисляют премию 1000 очков если в конце игры с поля убраны все квадраты.

Эта задача предоставляется с открытыми тестами: в качестве решения необходимо представить не программу, а выходные данные, соответствующие входным данным, находящимися на дискете. Входные данные на дискете записаны в файлах с именами вида CLICK?.SIS (на месте ? – код теста). Соответствующие выходные данные записать в файлы с именами вида CLICK?.VAL.

Входные данные. На первой строке текстового файла CLICK?.SIS записано число рядов N ($1 \leq N \leq 50$) и столбцов M ($1 \leq M \leq 50$). Следующие N строк описывают начальное положение игры – на каждой записана строка, состоящая из M целых чисел, где каждое число означает цвет одного квадрата.

Выходные данные. На первую строку текстового файла CLICK?.VAL вывести число сделанных игроком ходов K и на следующие K строк – ходы, сделанные игроком (на каждой строке – координаты квадрата, на котором был сделан очередной щелчок. Координаты левого нижнего угла поля – 1 1, правого верхнего угла – N M). На последнюю строку файла вывести число полученных к концу игры очков (включая премию).

Пример.	CLICKZ.SIS	CLICKZ.VAL
	4 8	6
	1 2 3 1 3 2 6 7	2 5
	2 3 5 5 5 7 6 7	1 3
	2 4 5 4 5 5 5 6	1 6
	2 3 5 3 5 6 7 6	1 6
		1 2
		1 1
		62

Оценивание. За каждый тест участник получает

0 баллов	если $S_L < 0,5 \cdot S_Z$
$X \cdot (2 \cdot S_L / S_Z - 1)$ баллов	если $0,5 \cdot S_Z \leq S_L \leq 1,25 \cdot S_Z$
$1,5 \cdot X$ баллов	если $S_L > 1,25 \cdot S_Z$

(X – стоимость теста, S_L и S_Z количество очков, набранных к концу игры соответственно решением участника и решением жюри).

Решение, в котором одним из ходов будет попытка щелкнуть на квадрате, не содержащемся в кластере, на пустом месте, или неверно указан итоговый результат игры, очков не получает.