

3. Playboard

You will be given a 3 by N playboard where each square has an integer between 1 and 1000 inclusively on it.

Three rules of the symbols are as below:

1. Every square MUST be marked with one of the symbols $\{\bigcirc, \triangle, \times\}$
2. Every column MUST contain all symbols.
3. There are no adjacent symbols that are same.

Once every square is marked with a symbol, then you are ready to calculate the total marks by adding numbers with the symbol \bigcirc , subtracting numbers with \times , and ignoring any numbers with the symbol \triangle .

[Case 1]

Given a 3 X 12 (N = 12) board is given as below:

2	8	11	15	9	28	19	16	41	34	28	9
10	13	9	16	20	18	32	26	15	37	24	3
5	6	7	16	25	31	21	29	39	29	19	10

Assume you have marked the squares as below:

\bigcirc	\triangle	\bigcirc	\times	\bigcirc	\times	\triangle	\bigcirc	\triangle	\bigcirc	\times	\bigcirc
\times	\bigcirc	\times	\triangle	\times	\triangle	\bigcirc	\times	\bigcirc	\times	\triangle	\times
\triangle	\times	\triangle	\bigcirc	\triangle	\bigcirc	\times	\triangle	\times	\triangle	\bigcirc	\triangle

Then you will be able to calculate the total marks with the given rules.

+2		+11	-15	+9	-28		+16		+34	-28	+9
-10	+13	-9		-20		+32	-26	+15	-37		-3
	-6		+16		+31	-21		-39		+19	

So the sum of all these is -35.

Write an algorithm which arranges $\{\bigcirc, \triangle, \times\}$ with regard to the rules and maximizes the total marks. N will not exceed 1,000,000.

As we are to exploit your algorithm with test cases with various size of N, the more efficient your algorithm is, the more marks you will achieve.

Again, you are not allowed to use optimization options at compilation and the runtime of your program processing the 10 cases should not exceed 1 second.

[Input]

An input file “input.txt” will be given, which has 10 test cases.

Each case is consist of four lines; first is length of the board ($1 \leq N \leq 1,000,000$), and the rest is the playboard row by row.

[Output]

For each case, you should print the case number as #x where x is the index of the case. Then, print the maximum marks you can get in the case.

You must produce your results as “output.txt”

[Example]

Input (input.txt)

10	← First Case
1 17 17 19 2 17 14 6 19 11	
1 19 16 6 13 15 16 13 7 2	
13 11 1 9 10 12 16 20 14 12	
30	← Second Case
35 45 22 15 26 15 19 38 2 42 15 34 20 4 40 28 23 16 26 15 10 36 35 30 34 6 49 22 24 11	
6 17 29 40 19 7 29 16 18 29 21 47 30 29 25 9 47 23 23 25 30 32 9 5 6 43 39 1 11 46	
35 12 46 26 32 24 37 44 15 26 47 2 27 12 40 50 45 11 16 6 45 43 45 14 8 12 49 21 3 44	
...	

Output (output.txt)

#1 75
#2 521
...