

3. 놀이판

길이가 N이고 3줄짜리 놀이판이 주어지고 놀이판의 각 칸에는 1부터 1000 자연수가 쓰여 있다. 이 놀이판의 모든 칸에 ○, △, ×를 써서 표시하는데, 세로 열마다 ○, △, ×를 각각 한 번씩 모두 사용해야 하고, 가로로 이웃하는 두 칸에는 같은 표시가 나타나서는 안된다. 이렇게 하면 ○, △, ×가 각각 n번씩 나타나게 된다.

모든 칸에 ○, △, ×를 표시하고 난 다음 이에 대해 점수를 매기는데, ○로 표시한 칸의 수는 더해주고, ×로 표기한 칸의 수는 빼주고, △로 표시한 칸의 수는 무시한다(0으로 처리). 아래 그림은 놀이판을 규칙대로 ○, △, × 표시한 예시이다.

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

○	△	○	×	○	×	△	○	△	○	×	○
×	○	×	△	×	△	○	×	○	×	△	×
△	×	△	○	△	○	×	△	×	△	○	△



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

첫 번째 표는 주어진 놀이판, 두 번째 표는 ○, △, ×를 채운 방법, 세 번째 표는 계산 규칙에 따라 더하는 수, 빼는 수, 무시하는 수를 나타내고 있다. 위의 방식으로 채울 때 점수는 $2 - 10 + 13 - 6 + \dots + 9 - 3 = 35$ 가 된다.

놀이판이 주어졌을 때 ○, △, ×를 표시하여 얻을 수 있는 최대 점수를 구하는 알고리즘을 작성하라. N은 1,000,000을 넘지 않는다. 여력 가지 크기의 문제로 테스트하므로 효율이 좋은 알고리즘일수록 점수를 많이 받을 것이며 원시적인 알고리즘으로는 큰 점수를 기대할 수 없다. 컴파일 시 최적화 옵션은 허용하지 않는다. 10개의 케이스를 모두 수행한 총 시간의 합이 1초를 넘지 않아야 한다.

[입력]

입력 파일에는 10개의 테스트 케이스가 주어진다. 각 케이스는 네 줄로 이루어진다. 첫 줄에는 N이 주어지고, 이후의 세 줄에는 차례대로 1, 2, 3행의 숫자들이 주어진다. 입력 파일의 이름은 “input.txt”이다.

[출력]

각 테스트 케이스에 대해서, 케이스의 번호를 “#x” 의 형식으로 출력한 후(여기서 x는 테스트 케이스 번호), 공백을 하나 둔 다음 주어진 케이스에서 받을 수 있는 최대 점수를 기록한다. 출력 결과물은 “output.txt”로 저장한다.

[예시]

입력 (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	
...	

출력 (output.txt)

#1 75
#2 521
...