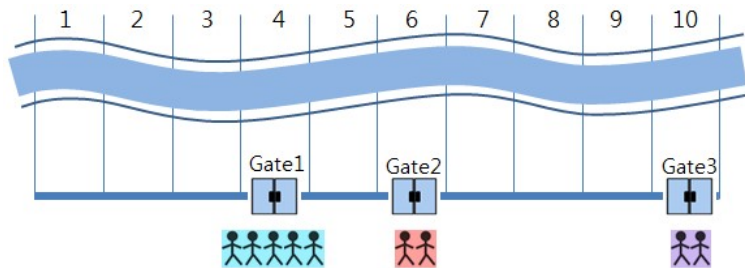


문제

문제명 : 낚시터 자리잡기

제한 조건	실행시간 : 50개의 테스트 케이스를 합쳐서 1초(C/C++/JAVA) 메 모 리 : Heap, Global, Stack 등을 모두 합해 최대 256MB까지 사용 가능 (단, 스택은 최대 1MB까지 사용 가능)
제출 횟수 제한	제한없음
채점	답안을 제출하면 결과를 판정해서실시간으로 알려주며 그 의미는 다음과 같다. Accept : Sample Input 에 대해 오류 없이 제출 Fail : 정상 제출되지 않았거나 Sample Input을 돌린 결과 0점 취득
평가	sample_input은 편의를 위해 제공하며, 실제 채점은 별도의 평가용 input으로 이루어진다. 제한 시간 이내에 평가용 input의 모든 테스트 케이스에 대해 정답을 도출하면 Pass 이며, 그 외에 오답, 부분 정답, 시간 제한을 초과하는 경우 1회 Fail이 된다.

강가를 따라서 낚시터 자리가 1~N까지 일렬로 늘어선다. 낚시터에는 3개의 출입구가 있으며, 각 출입구에는 낚시터에 입장하기 위해 대기하고 있는 낚시꾼들이 각각 존재한다.



[Fig. 1]

모든 낚시꾼들은 한 사람당 하나의 낚시터에 자리를 잡아야 하며, 자리를 잡는 절차는 다음과 같다.

- 혼잡을 피하기 위해 **하나의 출입구씩 선택하여 순차적으로 입장** 할 수 있다.
- 출입구가 선택되면, 해당 출입구에 대기하고 있는 낚시꾼들은 **자신의 위치에서 가장 가까운 빈 낚시터 자리로 한 명씩 이동하여 차례대로 자리를 잡는다.**
 - 출입구에서 바로 위쪽의 낚시터까지의 거리는 1m 이며, 좌우로 한 칸씩 멀어질 때 마다 추가로 1m씩 멀어진다.
 - 예를 들어 [Fig. 1]의 Gate1에서 4번 자리까지는 1m 이고, 3번과 5번자리는 2m의 거리가 된다.
- 해당 출입구의 맨 마지막 사람의 경우, 가장 가까운 빈 자리가 두 곳이라면 하나를 선택해야 한다.
(맨 마지막 사람이 아닌 경우, 두 곳 중 아무데나 가도 결과는 같으므로 고려할 필요가 없다.)
- 해당 출입구에 대기중인 모든 낚시꾼들의 자리잡기가 완료되면, 다음 출입구를 선택하여 위 1~3 과정을 반복 수행한다.

낚시터 자리의 개수 N이 주어지고, 출입구 3개의 위치 및 해당 출입구에 대기중인 각각의 낚시꾼들의 숫자가 주어진다.

이때 위의 낚시터 자리잡기 절차를 수행하면서 낚시꾼들 각각의 이동거리를 모두 더한 값이 최소가 되도록 자리잡는 방법을 찾고, 그때의 이동거리의 합을 출력하라.

예를 들어 앞의 [Fig. 1]과 같이, 낚시터 정보가 다음과 같이 주어졌을 경우를 살펴보자.

- 낚시터 자리 개수 N=10
- 1번 출입구의 위치=4, 낚시꾼들의 숫자=5명
- 2번 출입구의 위치=6, 낚시꾼들의 숫자=2명
- 3번 출입구의 위치=10, 낚시꾼들의 숫자=2명

[방법 1] Gate1 -> Gate2 -> Gate3 의 순서로 입장하는 경우.

이 경우 아래 [Fig. 2]와 같이 자리잡게 된다.