Practice Problem Set

Please check that you have 3 problems that are spanned across 7 pages in total (including Korean translation and this cover page).

Problem A
Balance Scale
Time Limit: 1 Second

There are N pebbles, numbered from 1 to n. There is a balance scale. We will put these pebbles on the scale according to the following rules.

1. Pebble 1 is put on the left pan and Pebble 2 is put on the right pan.
2. For Pebble $i = 3, \dots, n$, we do either A or B.
   A. If the scale is in equilibrium, Pebble i is put on the left pan.
   B. Otherwise, Pebble i is put on the lighter side.

After all the pebbles are put on the scale, the balance scale may not be in equilibrium. We will use additional masses for putting the scale in equilibrium. There are seven kinds of masses: 1g, 2g, 5g, 10g, 20g, 50g, and 100g. There is no limit to the number of masses of each kind.

Given the information on pebbles, write a program to output the smallest number of additional masses to put the scale in equilibrium in the end.

Input
Your program is to read from standard input. The input starts with a line containing an integer n ($2 \leq n \leq 10{,}000$), where n is the number of pebbles. The next line contains n integers where the i-th integer represents the weight of Pebble i. Each pebble weighs at least one and the sum of the weights of the pebbles is equal to or smaller than 10,000,000.

Output
Your program is to write to standard output. Print exactly one line. The line should contain the smallest number of additional masses to put the scale in equilibrium after the pebbles are put on the scale as described.

The following shows sample input and output for three test cases.

Sample Input 1	Output for the Sample Input 1
7 3 1 4 1 5 9 2	2

Sample Input 2	Output for the Sample Input 2
4 2 4 6 4	0

Sample Input 3	Output for the Sample Input 3
5 2 5 3 1 2	1

Problem A
양팔저울
Balance Scale
제한 시간: 1 초

1 부터 n까지 번호가 매겨진 n개의 자갈이 있다. 이 자갈들을 다음 절차에 따라 양팔저울에 올려놓는다.

1. 1 번 자갈을 왼쪽, 2 번 자갈을 오른쪽에 올려놓는다.
2. $i = 3, \ldots, n$번 자갈 각각에 대해서 차례로 다음 과정 중 하나를 수행한다.
   A. 만약 양팔저울이 평형을 이루는 경우, i번 자갈을 왼쪽에 올려 놓는다.
   B. 만약 양팔저울이 평형을 이루지 않는 경우, i번 자갈을 가벼운 쪽에 올려 놓는다.

모든 자갈을 위의 규칙에 따라 올려 놓은 후에도 양팔저울은 평형을 이루지 않을 수 있다. 이 경우 가벼운 쪽에 무게추를 올려서 균형을 맞추려고 한다. 무게추는 1g, 2g, 5g, 10g, 20g, 50g, 100g 7 종류가 있고, 무게추의 개수에는 제한이 없다.

입력 받은 자갈을 위 규칙에 따라 양팔저울에 올렸을 때, 최종적으로 평형을 맞추는데 추가적으로 필요한 무게추의 최소 개수를 구하는 프로그램을 작성하시오.

Input
입력은 표준입력을 사용한다. 첫 번째 줄에 자갈 개수를 나타내는 양의 정수 n $(2 \leq n \leq 10{,}000)$이 주어진다. 다음 줄에 n 개의 수들이 주어지는데, 이들은 번호 순서대로 자갈의 무게이다. 자갈의 무게는 각각 1 이상이며, 모든 자갈의 무게의 총합은 10,000,000 이하이다.

Output
출력은 표준출력을 사용한다. 최종적으로 평형을 맞추는데 추가적으로 필요한 무게추의 최소 개수를 한 줄에 출력한다.

다음은 세 테스트 케이스에 대한 입출력 예이다.

Sample Input 1	Output for the Sample Input 1
7 3 1 4 1 5 9 2	2

Sample Input 2	**Output for the Sample Input 2**
4 2 4 6 4	0

Sample Input 3	**Output for the Sample Input 3**
5 2 5 3 1 2	1

Problem B
Gift Discount
Time Limit: 1 Second

Given the prices of n gifts, we try to buy the maximum number of gifts with the budget of b. You write a program to find the maximum number of gifts with a budget b you can buy when you can get a half-price discount on up to a gifts. Note that you can only receive a half-price discount at most once per gift.

Input
Your program is to read from standard input. The input starts with a line containing three integers, $n (1 \leq n \leq 100{,}000)$ representing the number of gifts, $b (1 \leq b \leq 10^9)$ representing the budget, and $a (0 \leq a \leq n)$ representing the maximum number of gifts eligible for a half-price discount. The next line contains n integers representing the gift prices. You may assume that all gift prices are between 2 and 10^9 and are even numbers.

Output
Your program is to write to standard output. Print exactly one line. The line should contain the maximum number of gifts that can be purchased.

The following shows sample input and output for two test cases.

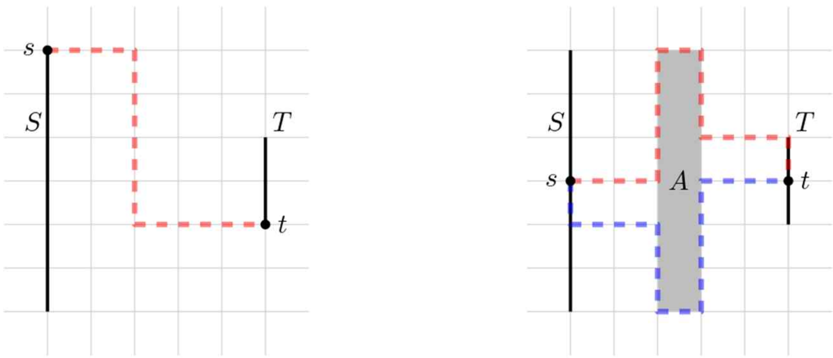
Sample Input 1	Output for the Sample Input 1
6 26 2 4 6 2 10 8 12	5

Sample Input 2	Output for the Sample Input 2
6 23 1 4 6 2 12 8 14	4

Problem C
Longest Shortest Paths
Time Limit: 2 Seconds

Consider n axis-aligned rectangles and two vertical segments S and T in the plane. We assume that all corners of the rectangles and segments are in integer coordinates. We also assume that the rectangles and segments are disjoint each other, that is, no two of them intersect each other and no two of them share a boundary point. For a point p in S and a point q in T, a *path* between p and q is a chain consisting of horizontal or vertical segments that connects p and q and does not intersect the interiors of the rectangles. The length of a path is the sum of the lengths of segments in the path. Thus, a shortest path between p and q is one whose length is the smallest among all paths between p and q.

For every pair of a point in S and a point in T, there is a shortest path between them. Let $d(p,q)$ denote the length of a shortest path between a point p in S and a point q in T. Our goal is to compute the length of the longest path among all shortest paths connecting a point in S and a point in T, that is, $\max_{p\in S}\max_{q\in T} d(p,q)$.



(a) (b)

For example, consider the figures above. Figure (a) shows two vertical segments S and T, and no rectangle in the plane. Every shortest path between a point in S and a point in T has length at most 9. Since $d(s,t) = 9$, we have $\max_{p\in S}\max_{q\in T} d(p,q) = d(s,t) = 9$ for this example.

Figure (b) shows an axis-aligned rectangle A and two vertical segments S and T in the plane. There are two shortest paths between a point s in S and a point t in T, one in red color and one in blue color. Then $d(s,t) = 11$. Observe that every shortest path between a point in S and a point in T has length smaller than or equal to 11. Thus, we have $\max_{p\in S}\max_{q\in T} d(p,q) = d(s,t) = 11$ for this example.

Given n axis-aligned rectangles and two vertical segments S and T that do not intersect each other, write a program to compute the length of the longest path among all shortest paths connecting a point in S and a point in T.

Input

Your program is to read from standard input. The input starts with a line containing six integers. The first three integers represent the x-coordinate and the two y-coordinates of the endpoints of the vertical segment S, and the last three integers represent the x-coordinate and the two y-coordinates of the endpoints of the vertical segment T.

The next line contains an integer n $(0 \leq n \leq 5{,}000)$, where n is the number of axis-aligned rectangles. The rectangles are numbered from 1 to n. In the following n lines, the i-th line contains four nonnegative integers. The first two integers represent the x-coordinate and y-coordinate of the top-left corner of the rectangle i, and the last two integers represent the x-coordinate and y-coordinate of the bottom-right corner of the rectangle i.

All the coordinate values of endpoints of S and T, and two corners of the rectangles are nonnegative integers no more than 10^8.

Output

Your program is to write to standard output. Print exactly one line. The line should contain the length of the longest path among all shortest paths between a point a point in S and a point in T.

The following shows sample input and output for three test cases. Sample input 1 corresponds to the case of Figure (a), and sample input 2 corresponds to the case of Figure (b).

Sample Input 1	Output for the Sample Input 1
0 0 6 5 2 4 0	9

Sample Input 2	Output for the Sample Input 2
0 0 6 5 2 4 1 2 6 3 0	11

Sample Input 3	Output for the Sample Input 3
0 10 30 5 10 30 2 2 50 3 12 2 11 3 0	41