

## 6764 Mario Kart

Have you ever played the Mario game? Of course you did, who did not?! Anyway, a new version of the Mario game has been released, it is some kind of kart racing game. And you decided to write a program to find the best strategy for you to complete each level.

Each level track can be modeled as an infinite straight line, with some stations at some specific points on this line. Each station has an integer, representing its position on the track. Your task is to go from the first station (the one with smallest position) to the last one (the one with largest position) in the minimum number of moves.

You can move between any two stations directly (you can go to a non-adjacent station, or you can go back to a station with a lower position if you want!) if you have enough boost coins for that move. In each level, you have some boost coins that you can use. Each boost coin has a cost and power value. You can select any subset of the coins for each move, but each coin may be used only once per move. Coins are permanent, and you can use the coins again for other moves in the same level.

To make a move, you must choose a subset of the boost coins, such that the sum of their costs must not exceed  $L$ , and the sum of their power values must be exactly equal to the absolute difference between the positions of the two stations you are moving between. If there is no such subset, then you cannot make that move directly.

Now you are given some configurations for some levels, and you are required to find the minimum number of moves to finish each one, or say it is impossible to finish that level.

### Input

Your program will be tested on one or more test cases. The first line of the input will be a single integer  $T$ , the number of test cases ( $1 \leq T \leq 100$ ). Followed by the test cases, the first line of each test case contains 3 integers separated by a single space  $N M L$  ( $2 \leq N \leq 100$ ), ( $1 \leq M \leq 100$ ) and ( $1 \leq L \leq 1,000$ ) representing the number of stations, the number of boost coins and the maximum sum of coins' costs in each move, respectively. Followed by a line which contains  $N$  unique positive integers separated by a single space (not necessary sorted, and each integer will be at most 1,000), representing the positions of the stations. Followed by  $M$  lines, each line contains 2 integers separated by a single space  $C V$  ( $1 \leq C, V \leq 100$ ) which represent the cost and the power value of a coin, respectively.

### Output

For each test case, print a single line which contains a single integer, this integer should be '-1' if there is no way to go from the left most station to the right most station, or the minimum number of moves to do it if it is possible.

### Note

In the first test case, the station positions are [3, 1, 6], and you start at 1 and must end at 6. You will have to make 2 moves, going from  $1 \rightarrow 3$  using the coin (3, 2), and going from  $3 \rightarrow 6$  using the coin (3, 3). You can not go directly from  $1 \rightarrow 6$  using (3, 2) and (3, 3) because the sum of the costs of the coins exceeds the limit 4.

### Sample Input

```
2
3 2 4
```

3 1 6  
3 2  
3 3  
3 1 4  
1 3 6  
3 2

### Sample Output

2  
-1