# Phone Lines

There are several cities and towers on a straight line. Towers can be set to connection-accepting by paying a cost. We are given the location (on the X-axis), of the towers and the cities. Our job is to set up certain towers as connection-accepting. Now every city, pays you an amount equal to **D - distance_travelled_by_data**, for every unit of data (for every tower) it can send. (distance_travelled_by_data = cityX - towerX); Our job here is to setup connections on differrent towers to get maximal profit.
Each city when it wants to route some data to a tower works with the following algorithm:
(1) Find the nearest tower to the left of the city.
(2) If it is within the range 'D', it sends the data to that tower. If this tower exceeds the range D, or if the tower doesnt accept connections, the city cant send the data and stops immediately. (Doesnt check the next available tower);
(3) If the data is sent sucessfully: Then the city
(3.1) Skips three towers. (Doesnt care if these three towers are connection-accepting or not);
(3.2) Tries to send data to the next tower (the fouth one after the skipping), by using step (2);

Input format:
Input consists of multiple testcases.
First line of each test case, contains two integers: D C T; The range, the number of cities and the number of towers, respectively.
Second line contains exactly C integers saying the location of the cities (on the X-axis).
The next T lines contain exactly two integers: location[i] connection-cost[i]; which is the position of tower i, and the cost to setup tower i as connection-accepting;
The input ends with a line: "-1 -1 -1"

Output format:
For each test case, output a single line saying the maximum amount of profit you can make.

Constraints:
Now two points (towers or cities), will have the same X-coordinate. T,C <= 100.

Sample Input:

```
4 9 6
23
43
18
15
29
50
41
31
40
32 2
26 0
46 7
48 0
50 3
38 1
-1 -1 -1
```

Sample Output:

5