Assignment 3

Part 1: Dynamic Programming

Question 1 [10 Marks]

It's commonly known that the Dutch have invented copper-wire. Two Dutch men were fighting over a nickel, which was made of copper. They were both so eager to get it and the fighting was so fierce, they stretched the coin to great length and thus created copper-wire.
Not commonly known is that the fighting started, after the two Dutch tried to divide a bag with coins between the two of them. The contents of the bag appeared not to be equally divisible. The Dutch of the past couldn't stand the fact that a division should favour one of them and they always wanted a fair share to the very last cent. Nowadays fighting over a single cent will not be seen anymore, but being capable of making an equal division as fair as possible is something that will remain important forever…
That's what this whole problem is about. Not everyone is capable of seeing instantly what's the most fair division of a bag of coins between two persons. Your help is asked to solve this problem.
Given a bag with a maximum of 100 coins, determine the most fair division between two persons. This means that the difference between the amount each person obtains should be minimised. The value of a coin varies from 1 cent to 500 cents. It's not allowed to split a single coin.

Input

A line with the number of problems n, followed by n times:

- A line with a non negative integer m (m ≤ 100) indicating the number of coins in the bag
- A line with m numbers separated by one space, each number indicates the value of a coin.

Output
The output consists of n lines. Each line contains the minimal positive difference between the amount the two persons obtain when they divide the coins from the corresponding bag.

Sample

Input	Output

2
3
2 3 5
4
1 2 4 6
	0
	1

Question 2 [10 Marks]

Our friends at TXT have a bug-ridden message sender.
Specifically, this sender can send a string s to TXT or others, but if the character “w” is entered, it will send “uu” instead of “w”, and if the character “m” is entered, it will send “nn” instead of “m”.
That day TXT received a message from his friend, and he knew the message was wrong.
Now please can you help TXT calculate the number of the possibilities of the original string s

Input
The input consists of a line containing a string s contains only lowercase Latin letters. (1 ≤ | s | ≤ 10^5)

Output
Print an integer - the original string s number of possibilities, modulo 10^9+7.

Sample

Input	Output
ouuoharinn banana nnn amanda	3 1 3 0

Explanation

For the first example, the candidate strings are as follows: “ouuoharinn”, “ouuoharim”, “owoharim”, and “owoharinn”.
For the second example, there is only one: “banana”.
For the third example, the candidate strings are as follows: “nm”, “mn” and “nnn”.
For the last example, no candidate string can be converted to “amanda” because the message sender will not send “m”.

Question 3 [10 Marks]

Danilo Gheyi is a renowned bank robber. He is known worldwide for accomplishing the most profitable bank robbery, in Fortaleza, Ceará. Danilo and his friends dug a tunnel to get into the main chest. There were some bags, with different amounts of money or jewelry and weight. They had to leave about 50% of the total value, since the truck couldn't carry all the bags.

Danilo wasn't caught, and to show that he can do it all again, he is planning a robbery at one of the safer banks in the USA -the Wachovia Bank. He wants your help to maximize the amount stolen, avoiding a huge loss as it happened in Fortaleza.

Write a program that, given the maximum weight the truck is able to carry and the information about each bag in the bank, determine the maximum value that Danilo can steal.

Input
The input consists of several instances. There is an integer N (1 ≤ N ≤ 200) in the first line; it stands for the number of instances. The first line of each instance contains two integers, K and M (8 ≤ K ≤ 1000 and 1 ≤ M ≤ 50) representing, respectively, the maximum weight the truck can handle and the amount of bags in the bank. The next M lines describe each bag with two integers A and B (8 ≤ A ≤ 200 and 1 ≤ B ≤ 25): the weight and the value of the bag, respectively.

Output
For each instance output a sentence “Hey stupid robber, you can get P.”, and P represents the maximum value Danilo can steal.

Sample

Input	Output

3
34 5
178 12
30 1
13 7
34 8
87 6
900 1
900 25
100 10
27 16
131 9
132 17
6 5
6 23
56 21
100 25
1 25
25 25
100 2
	Hey stupid robber, you can get 8.
	Hey stupid robber, you can get 25.
	Hey stupid robber, you can get 99.

Question 4 [10 Marks]

John is a diver and a treasure hunter. He has just found the location of a pirate ship full of treasures. The sophisticated sonar system on board his ship allows him to identify the location, depth and quantity of gold in each sunken treasure. Unfortunately, John forgot to bring a GPS device and the chances of ever finding this location again are very slim so he has to grab the gold now. To make the situation worse, John has only one compressed air bottle.

John wants to dive with the compressed air bottle to recover as much gold as possible from the wreck. Write a program John can use to select which treasures he should pick to maximize the quantity of gold recovered.

The problem has the following restrictions:

- There are n treasures {(d1, v1),(d2, v2), . . .(dn, vn)} represented by the pair (depth, quantity of gold). There are at most 30 treasures.
- The air bottle only allows for t seconds under water. t is at most 1000 seconds.
- In each dive, John can bring the maximum of one treasure at a time.
- The descent time is tdi = w ∗ di , where w is an integer constant.
- The ascent time is tai = 2w ∗ di , where w is an integer constant.
- Due to instrument limitations, all parameters are integer.

Input
The input to this program consists of a sequence of integer values. Input contains several test cases. The first line of each dataset should contain the values t and w. The second line contains the number of treasures. Each of the following lines contains the depth di and quantity of gold vi of a different treasure. A blank line separates each test case.

Note: In this sample input, the bottle of compressed air has a capacity of 200 seconds, the constant w has the value 4 and there are 3 treasures, the first one at a depth of 10 meters and worth 5 coins of gold, the second one at a depth of 10 meters and worth 1 coin of gold, and the third one at 7 meters and worth 2 coins of gold.

Output

The first line of the output for each dataset should contain the maximum amount of gold that John can recover from the wreck. The second line should contain the number of recovered treasures. Each of the following lines should contain the depth and amount of gold of each recovered treasure. Treasures should be presented in the same order as the input file. Print a blank line between outputs for different datasets.

Sample

Input	Output
210 4	7
3	2
10 5	10 5
10 1	7 2
7 2	

Question 5 [10 Marks]

Given a list of numbers A, find the length of the longest increasing subsequence in it. A subsequence is a sequence that can be derived from the given sequence by deleting zero or more elements without changing the order of the remaining elements. An increasing subsequence is a subsequence in which each element is greater than all previous elements.
For example, in the list {33 , 11 , 22 , 44}:
The subsequences {33 , 44} and {11} are increasing subsequences while {11 , 22 , 44} is the longest increasing subsequence.

Input
First line contains a single integer N (1 <= N <= 10) the length of the list A.
The second line contains N numbers (1 <= each number <= 20), the numbers in the list A separated by spaces.

Output
One line containing the length of the longest increasing subsequence in A.

Sample

Input	Output
5 1 4 2 4 3	3

Part 2: Greedy algorithms and MST

Question 1 [10 Marks]

Assume you are an awesome parent and want to give your children some cookies. But, you should give each child at most one cookie.
Each child i has a greed factor g[i], which is the minimum size of a cookie that the child will be content with; and each cookie j has a size s[j]. If s[j] >= g[i], we can assign the cookie j to the child i, and the child i will be content. Your goal is to maximize the number of your content children and output the maximum number.

Example 1:
Input: g = [1,2,3], s = [1,1]
Output: 1
Explanation: You have 3 children and 2 cookies. The greed factors of 3 children are 1, 2, 3. And even though you have 2 cookies, since their size is both 1, you could only make the child whose greed factor is 1 content.
You need to output 1.

Example 2:
Input: g = [1,2], s = [1,2,3]
Output: 2
Explanation: You have 2 children and 3 cookies. The greed factors of 2 children are 1, 2. You have 3 cookies and their sizes are big enough to gratify all of the children,
You need to output 2.

Constraints:

- 1 <= g.length <= 3 * 104
- 0 <= s.length <= 3 * 104
- 1 <= g[i], s[j] <= 231 - 1

Question 2 [10 Marks]

You are given an array of CPU tasks, each labeled with a letter from A to Z, and a number n. Each CPU interval can be idle or allow the completion of one task. Tasks can be completed in any order, but there's a constraint: there has to be a gap of at least n intervals between two tasks with the same label.
Return the minimum number of CPU intervals required to complete all tasks.

Example 1:
Input: tasks = ["A","A","A","B","B","B"], n = 2
Output: 8
Explanation: A possible sequence is: A -> B -> idle -> A -> B -> idle -> A -> B.
After completing task A, you must wait two intervals before doing A again. The same applies to task B. In the 3rd interval, neither A nor B can be done, so you idle. By the 4th interval, you can do A again as 2 intervals have passed.

Example 2:
Input: tasks = ["A","C","A","B","D","B"], n = 1
Output: 6
Explanation: A possible sequence is: A -> B -> C -> D -> A -> B.
With a cooling interval of 1, you can repeat a task after just one other task.

Example 3:
Input: tasks = ["A","A","A", "B","B","B"], n = 3
Output: 10
Explanation: A possible sequence is: A -> B -> idle -> idle -> A -> B -> idle -> idle -> A -> B.
There are only two types of tasks, A and B, which need to be separated by 3 intervals. This leads to idling twice between repetitions of these tasks.

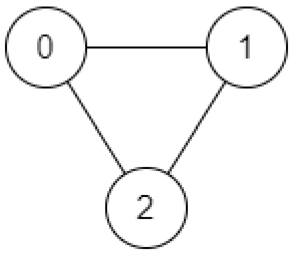
Constraints:

- 1 <= tasks.length <= 104
- tasks[i] is an uppercase English letter.
- 0 <= n <= 100

Question 3 [10 Marks]

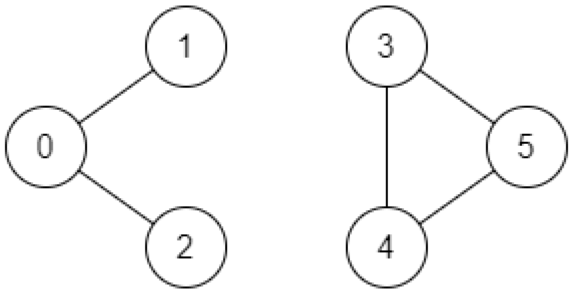
There is a bi-directional graph with n vertices, where each vertex is labeled from 0 to n - 1 (inclusive). The edges in the graph are represented as a 2D integer array edges, where each edges[i] = [ui, vi] denotes a bi-directional edge between vertex ui and vertex vi. Every vertex pair is connected by at most one edge, and no vertex has an edge to itself.
You want to determine if there is a valid path that exists from vertex source to vertex destination.
Given edges and the integers n, source, and destination, return true if there is a valid path from source to destination, or false otherwise.

Example 1:



Input: n = 3, edges = [[0,1],[1,2],[2,0]], source = 0, destination = 2
Output: true
Explanation: There are two paths from vertex 0 to vertex 2:
- 0 → 1 → 2
- 0 → 2

Example 2:



Input: n = 6, edges = [[0,1],[0,2],[3,5],[5,4],[4,3]], source = 0, destination = 5
Output: false
Explanation: There is no path from vertex 0 to vertex 5.

Constraints:

- 1 <= n <= 2 * 105
- 0 <= edges.length <= 2 * 105
- edges[i].length == 2
- 0 <= ui, vi <= n - 1
- ui != vi
- 0 <= source, destination <= n - 1
- There are no duplicate edges.
- There are no self edges.

Question 4 [10 Marks]

The kingdom of Olympia consists of N cities and M bidirectional roads. Each road connects exactly two cities and two cities can be connected with more than one road. Also it possible that some roads connect city with itself making a loop.

All roads are constantly plundered with bandits. After a while bandits became bored of wasting time in road robberies, so they suggested the king of Olympia to pay off. According to the offer, bandits want to get a gift consisted of gold and silver coins. Offer also contains a list of restrictions: for each road it is known gi — the smallest amount of gold and si — the smallest amount of silver coins that should be in the gift to stop robberies on the road. That is, if the gift contains a gold and b silver coins, then bandits will stop robberies on all the roads that gi ≤ a and si ≤ b.

Unfortunately kingdom treasury doesn't contain neither gold nor silver coins, but there are Olympian tugriks in it. The cost of one gold coin in tugriks is G, and the cost of one silver coin in tugriks is S. King really wants to send bandits such gift that for any two cities there will exist a safe path between them. Your task is to find the minimal cost in Olympian tugriks of the required gift.

Input
The first line of the input contains two integers N and M (2 ≤ N ≤ 200, 1 ≤ M ≤ 50 000) — the number of cities and the number of roads, respectively. The second line contains two integers G and S (1 ≤ G, S ≤ 109) — the prices of gold and silver coins in tugriks. The following M lines contain information about the offer. Each of the records in list is given as four integers xi, yi, gi, si, where xi and yi are the numbers of cities that the road connects and gi, si are minimal gold and silver coins requirements for the i-th road (1 ≤ xi, yi ≤ N, 1 ≤ gi, si ≤ 109). Cities are numbered from 1 to N. It is possible that there are more than one road between a pair of cities. It is possible that a road connects the city with itself.

Output
The output should contain the minimal cost of the gift in Olympian tugriks. If there is no gift that satisfies the given requirements output -1 .

Example Input
```
3 3
2 1
1 2 10 15
1 2 4 20
1 3 5 1
```

Example Output
```
30
```

Question 5 [10 Marks]

John lives in HackerLand, a country with N cities and M bidirectional roads. Each of the roads has a distinct length, and each length is a power of two (i.e., 2 raised to some exponent). It's possible for John to reach any city from any other city.

Given a map of HackerLand, can you help John determine the sum of the minimum distances between each pair of cities? Print your answer in binary representation.

Input Format

The first line contains two space-separated integers denoting N (the number of cities) and M (the number of roads), respectively.
Each line of the subsequent lines contains the respective values of Ai, Bi, and Ci as three space-separated integers. These values define a bidirectional road between cities Ai and Bi having length 2^Ci.

Constraints

1 <= N <= 100000
1 <= M <= 2 * 100000
1 <= Ai, Bi <= N, Ai != Bi
0 <= Ci < M
if i != j, then Ci != Cj

Output Format

Find the sum of minimum distances of each pair of cities and print the answer in binary representation.

Sample Input

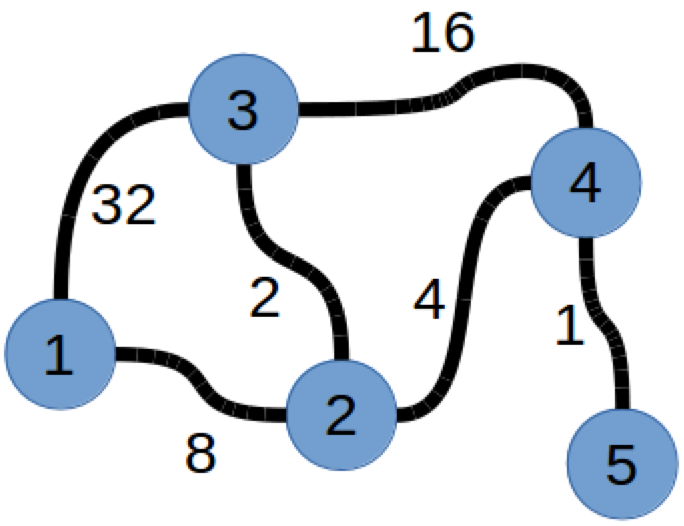
```
5 6
1 3 5
4 5 0
2 1 3
3 2 1
4 3 4
4 2 2
```

Sample Output

```
1000100
```

Explanation

In the sample, the country looks like this:



Let $d(x, y)$ be the minimum distance between city x and city y.

$d(1, 2) = 8$
$d(1, 3) = 10$
$d(1, 4) = 12$
$d(1, 5) = 13$
$d(2, 3) = 2$
$d(2, 4) = 4$
$d(2, 5) = 5$
$d(3, 4) = 6$
$d(3, 5) = 7$
$d(4, 5) = 1$

$Sum = 8 + 10 + 12 + 13 + 2 + 4 + 5 + 6 + 7 + 1 = (68)_{10} = (1000100)_2$

Submission Guidelines:

- You should work in teams of 4
- No late submissions are allowed.
- Cheating is NOT tolerated by any means.
- TAs will grade the assignment out of 100, but this score may be adjusted later for scaling purposes.

Deliverables:

- ONLY the team leader should submit a Zip file under the name: ***<G#_TeamLeaderName_TeamLeaderID>***
- The zip file should include:
    - A text file with the teams' names and IDs
    - A CPP file for each question containing the code.

Deadline:

- Date: Tuesday, 31th of December
- Time: 11:59 PM