**JOI 2013-2014 Qualification contest**

Authors: JCIOI (Japanese Committee for the IOI)

Translators: Jasmine Chua, France-ioi

License: Creative Commons Attributions-ShareAlike 4.0 International License (CC BY-SA 4.0)

**Average Score**

**QUESTION**

Five students, Taro, Jiro, Saburo, Shiro, and Hanako, were present during a lecture at JOI High School.

A semester exam was administered during this lecture, and all 5 students sat for the exam. Students who scored ≥40 points in the exam will have that as their final score. Students who scored <40 points will go for remedial class and get 40 points as their final score.

Given the points scored by the 5 students, create a program to calculate the average score.

**INPUT**

The input has 5 rows, with 1 integer written in each row.

In the first row, Taro's score for the semester exam is written.

In the second row, Jiro's score for the semester exam is written.

In the third row, Saburo's score for the semester exam is written.

In the fourth row, Shiro's score for the semester exam is written.

In the fifth row, Hanako's score for the semester exam is written.

All 5 students' scores are between 0 and 100.

All 5 students' scores are multiples of 5.

The average score of the 5 students must be an integer.

**OUTPUT**

Output the average score of the 5 students in one row.

**INPUT/OUTPUT EXAMPLES**

**INPUT 1**

10

65

100

30

95

**OUTPUT 1**

68

**INPUT 2**

40

95

0

95

50

**OUTPUT 2**

64

In INPUT EXAMPLE 1, Taro and Shiro's scores are <40 points, so their final score is 40 points each. Jiro, Saburo, and Hanako's scores are ≥40 points, so their final scores are 65 for Jiro, 100 for Saburo, and 95 for Hanako. The total score of the 5 students is 340, so the average score is 68.

※Save each example's input/output data as a file by right-clicking the links.

**Vote**

**QUESTION**



In 20XX, Tokyo will hold an international sporting event. To introduce programing as a sport to the world, it was shortlisted as one of the competitions that may be held during the event. The judging committee that decides on sport selection did some research, and conducted the following steps:

* The N sports that are shortlisted were compiled into a list and ranked according to how interesting it is. The sport in the ith row of the list is deemed to be the ith most interesting sport, and is termed Sport i. In addition, the cost needed to hold a Sport i competition is written as Ai.
* The judging committee is made up of M members, named Member 1 to Member M. Member j has their personal judgement criteria Bj. Judgement criteria Bj will be used to weed out sports that have costs that exceed Bj, then the member will cast 1 vote in favor of the sport ranked as most interesting among the remaining sports.
* The judgement criteria for every member ensures that there will be at least 1 sport that meets his criteria. Therefore, each member can cast 1 vote.
* There will only be 1 sport that gets the most number of votes.

Given the list and information about the committee members, create a program to find out the number of the sport that received the most number of votes.

**INPUT**

The input has (1 + N + M) rows.

In the first row, two integers are written.

* N, where 1 ≦ N ≦ 1000. This represents the number of sports on the list.
* M, where 1 ≦ M ≦ 1000. This represents the number of committee members.

For the next N rows, in the ith row, where 1 ≦ i ≦ N, one integer is written.

* Ai, where 1 ≦ Ai ≦ 1000. This represents the cost needed to hold a Sport i competition.

For the following M rows, in the jth row, where 1 ≦ j ≦ M, one integer is written.

* Bj, where 1 ≦ Bj ≦ 1000. This represents Member j's judgement criteria.

For the given input data, ensure that each member casts 1 vote, and that there is only 1 sport that gets the most number of votes.

**OUTPUT**

Output the number of the sport that received the most number of votes in one row.

**INPUT/OUTPUT EXAMPLES**

**INPUT 1**

4 3

5

3

1

4

4

3

2

**OUTPUT 1**

2

**INPUT 2**

6 6

3

1

4

1

5

9

2

6

5

3

5

9

**OUTPUT 2**

1

In INPUT EXAMPLE 1, there are 4 sports and 3 committee members. The costs needed for the 4 sports in the list are 5, 3, 1, and 4 respectively.

* Member 1's judgement criteria is 4. In the list of sports with costs ≤4, Sport 2 is the most interesting.
* Member 2's judgement criteria is 3. In the list of sports with costs ≤3, Sport 2 is the most interesting.
* Member 3's judgement criteria is 2. In the list of sports with costs ≤2, Sport 3 is the most interesting.

Therefore, Sport 2 gets 2 votes and Sport 3 gets 1 vote. Sport 2 is the one which received the most number of votes, so 2 is the output.

In INPUT EXAMPLE 2, Sport 1 gets 5 votes and Sport 2 gets 1 vote. Sport 1 is the one which received the most number of votes, so 1 is the output.

※Save each example's input/output data as a file by right-clicking the links.

**Super Metropolis**

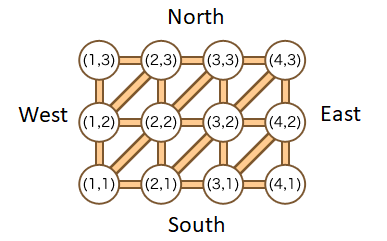
**QUESTION**

JOI is creating a sightseeing tour itinerary of super metropolis IOI Kingdom.

The super metropolis has W roads stretching from North to South, and H roads stretching from West to East, dividing the kingdom up into a grid.

The W roads stretching from North to South are labeled 1, 2, ... , W starting from the West. The H roads stretching from West to East are labeled 1, 2, ... , H from the South. The ith road from the West and jth road from the South meet at intersection (i, j).

Also, as in the figure below, each intersection has a road that extends in the North-east direction (except the northernmost and easternmost intersections). They also have a road that extends in the South-west direction (except the southernmost and westernmost intersections). In other words, intersection (i, j) has roads that connect it to intersections (i - 1, j), (i + 1, j), (i, j - 1), and (i, j + 1). In addition, there may also be roads connecting it to intersections (i - 1, j - 1) and (i + 1, j + 1).



JOI is deciding the order to visit N sightseeing spots. The ith sightseeing spot, where 1 ≦ i ≦ N, is located at intersection (Xi, Yi). JOI hopes to make the tour as short as possible by traveling on as few roads as he can. Create a program to find out the minimum total number of roads he must travel on to visit the sightseeing spots in a predetermined order.

The tour starts at intersection (X1, Y1). The tour cannot travel out of the metropolis. JOI can pass through an intersection without visiting the sightseeing spot located there.

**(Added after preliminary competition)** Extra note regarding 'total number of roads': JOI can travel on the same road more than once during the tour. In this case, the 'total number of roads' will include each passing separately.

**INPUT**

The input has (1 + N) rows.

In the first row, three integers are written with a space between them.

* W, where 2 ≦ W ≦ 10000.
* H, where 2 ≦ H ≦ 10000.
* N, where 1 ≦ N ≦ 1000.

For the next N rows, in the ith row, where 1 ≦ i ≦ N, two integers are written with a space between them.

* Xi, where 1 ≦ Xi ≦ W.
* Yi, where 1 ≦ Yi ≦ H. These represent that the ith sightseeing spot is located at intersection (Xi, Yi).

**OUTPUT**

Output the minimum total number of roads that must be travelled on to visit the sightseeing spots in a predetermined order in one row.

**INPUT/OUTPUT EXAMPLES**

**INPUT 1**

4 3 3

1 1

3 3

4 1

**OUTPUT 1**

3

**INPUT 2**

4 3 5

1 3

4 3

2 2

2 2

1 3

**OUTPUT 2**

7

In INPUT EXAMPLE 1, the best route is to move in the order: (1, 1), (2, 2), (3, 3), (3, 2), (4, 2), (4, 1).

In INPUT EXAMPLE 2, JOI passes through some intersections more than once.

※Save each example's input/output data as a file by right-clicking the links.

**Schedule**

**QUESTION**

There are 3 members in IOI High School's programing club, students J, O, and I. They are working to create a club activity schedule.

They want to create a schedule with N days of activities. There is only 1 club room key, which is currently in student J's possession. On each day, the key must be in the possession of any of the students that are attending club activities on that day. The key can then can then be brought home by any of the students present on that day.

The programming club must have activities on each day, and the student in-charge for each day is determined in advance. That student in-charge must attend club activities for that respective day.

Given the number of days of activities and the student in-charge for each day, create a program to find out the number of possible schedules, with the output as the remainder of the number of possibilities divided by 10007. Take note that the key must be brought home by a member that attended club activities for that day. Any member can bring home the key on the final day.

**INPUT**

The input has 2 rows.

In the first row, one integer is written.

* N, where 2 ≦ N ≦ 1000. This represents the number of days of activity on the schedule.

In the second row, a string of N characters is written. The ith character, where 1 ≦ i ≦ N, represents the student in-charge of each day. In other words, the characters J, O, or I is written to represent students J, O, or I respectively.

**OUTPUT**

Output the number of possible schedules, with the output as the remainder of the number of possibilities divided by 10007 in one row.

**INPUT/OUTPUT EXAMPLES**

**INPUT 1**

2

OI

**OUTPUT 1**

7

**INPUT 2**

20

JIOIJOIJOJOIIIOJIOII

**OUTPUT 2**

4976

In INPUT EXAMPLE 1, the schedule covers 2 days of club activities. The students in-charge are student O on day 1 and student I on day 2. There are 7 possible schedule that fulfil the question's conditions.

|  |  |  |
| --- | --- | --- |
|  | Day 1 | Day 2 |
| Schedule 1 | J, O | O, I |
| Schedule 2 | J, O | J, I |
| Schedule 3 | J, O | J, O, I |
| Schedule 4 | J, O, I | I |
| Schedule 5 | J, O, I | J, I |
| Schedule 6 | J, O, I | O, I |
| Schedule 7 | J, O, I | J, O, I |

The schedules represent which students will attend each day's activities. Take note that while student O is in-charge on day 1, the key is in student J's possession, so both students J and O must attend club activities on day 1. Also, the student who brings home the key on day 1 must attend activities on day 2, so ensure that there is at least 1 student who attends activities on both days.

In INPUT EXAMPLE 2, there are 72493594992 possible schedules that fulfil the question's conditions. Dividing that by 10007 will give an output on 4976.

※Save each example's input/output data as a file by right-clicking the links.

**Taxis**

**QUESTION**

There are N towns in IOI Kingdom, named Town 1 to Town N. The towns are connected by roads. There are K roads in IOI Kingdom, with each road connecting 2 different towns. Cars are free to move in either direction along the roads, but there is no other way to travel between towns.

JOI lives in Town 1 of IOI Kingdom, and is taking a taxi to visit his grandmother, who lives in Town N. There are N taxi companies in IOI Kingdom, named Company 1 to Company N. The taxi companies operate via the following special rules:

* JOI can only get on Company i's taxi in Town i.
* Company i's fare depends on Ci, the distance travelled.
* Company i's taxi can travel along a maximum of Ri roads.

For example, if R1 = 2, JOI will get on Company 1's taxi in Town 1, and travel along a maximum of 2 roads. If he has to travel along more than 2 roads, he has to switch taxis along the way.

JOI can only exit a taxi when it reaches a town. In addition, he can only travel by taxi. Create a program to find out the minimum total fare JOI must pay to reach Town N.

**INPUT**

The input has (1 + N + K) rows.

In the first row, two integers are written with a space between them.

* N, where 2 ≦ N ≦ 5000. This represents that there are N towns in IOI Kingdom.
* K, where N - 1 ≦ K ≦ 10000. This represents that there are K roads in IOI Kingdom.

For the next N rows, in the ith row, where 1 ≦ i ≦ N, two integers are written with a space between them.

* Ci, where 1 ≦ Ci ≦ 10000. This represents that the fare for Company i is Ci.
* Ri, where 1 ≦ Ri ≦ N. This represents that Company i's taxi can travel along a maximum of Ri roads.

For the next K rows, in the jth row, where 1 ≦ j ≦ K, two integers are written with a space between them.

* Aj and Bj, where 1 ≦ Aj ＜ Bj ≦ N. This represents that there is a road between Towns A and B. The combination of (Aj, Bj) can only be written once.

For the given data, ensure that JOI can switch taxis at any town to head to another town.

**OUTPUT**

Output the minimum total fare JOI must pay to reach Town N in one row.

**INPUT/OUTPUT EXAMPLES**

**INPUT**

6 6

400 2

200 1

600 3

1000 1

300 5

700 4

1 2

2 3

3 6

4 6

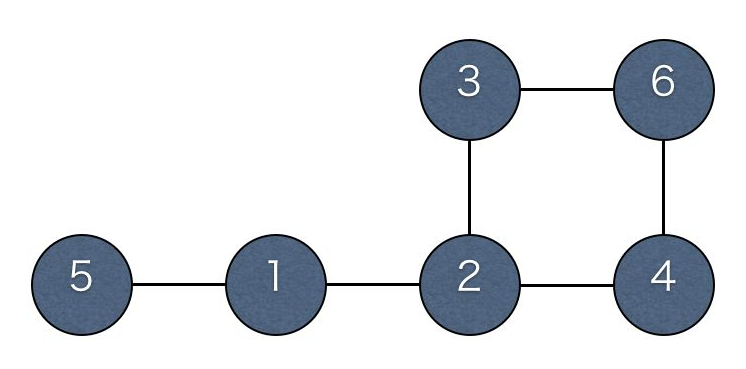
1 5

2 4

**OUTPUT**

700

The INPUT EXAMPLE above is represented in the following figure, with towns as circles and roads as lines.



To travel to Town 6 on a minimum total fare, JOI will do the following:

* Move from Town 1 to Town 5. (Fare=400)
* Move from Town 5 to Town 6. (Fare=300)

By using this route, the total fare is 400 + 300 = 700. This is the minimum, so the output is 700.

※Save each example's input/output data as a file by right-clicking the links.

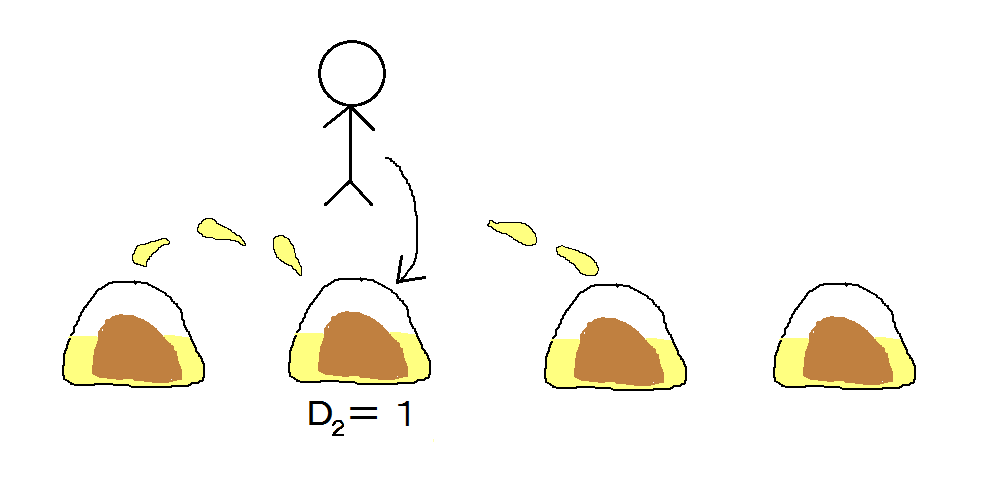
**Chinese Dumplings**

**QUESTION**

JOI is having a lunch of Chinese dumplings at an eatery. Chinese dumplings consist of meat and hot soup wrapped in a flour skin and steamed. Upon biting into it, the soup bursts out from the skin and scatters in all directions.

The set JOI ordered has N dumplings, each with a different amount of meat and soup. The N dumplings are lined up in one row, and labeled from 1 to N. The ith dumpling and jth dumpling are separated by a distance of |i - j|.

JOI starts eating the dumplings in a certain order. At first, all dumplings have a flavor rating of 0. When he eats dumpling i, the soup from that dumpling scatters and lands on any dumplings within a distance ≤Di, the distance from dumpling i. Dumplings that are covered in soup gains a flavor rating of Ai. In other words, when eating dumpling i, dumpling j is still on the table, where 1 ≦ j ≦ N and i - Di ≦ j ≦ i + Di, so dumpling j's flavor rating will increase by Ai.



JOI wants to decide on the order to eat the dumplings to maximize the total flavor rating. Create a program to find out the maximum total flavor rating of the dumplings.

**INPUT**

The input has 3 rows.

In the first row, one integer N, where 1 ≦ N ≦ 100, is written.

In the second row, a string with N integers, D1, D2, ..., DN, where 0 ≦ Di ≦ 7, is written with a space between them.

In the third row, a string with N integers, A1, A2, ..., AN, where 0 ≦ Ai ≦ 1000, is written with a space between them.

**OUTPUT**

Output the maximum total flavor rating of the dumplings in one row.

**INPUT/OUTPUT EXAMPLES**

**INPUT 1**

5

1 0 1 1 2

0 2 6 3 4

**OUTPUT 1**

20

**INPUT 2**

10

5 2 7 2 6 5 3 5 3 6

8 7 8 4 0 6 0 10 10 0

**OUTPUT 2**

237

In INPUT EXAMPLE 1, if JOI eats the dumplings in this order: 5 → 3 → 1 → 2 → 4, the total flavor rating is 20. Since this is the maximum flavor rating achievable, this is the best way of eating the dumplings.

※Save each example's input/output data as a file by right-clicking the links.