TWGSS OI Team

2024/25 TEAM FORMATION TEST

SCHEDULE

Date	28 September 2024
Time	08:45 - 11:45 | 3 hours

PROBLEMS

	Problem	*Limit*		*Score*	
		Time	*Memory*	*Subtasks*	*Total*
TF251	Standard Charter	1s	256MB	5 | 9 | 11 | 15	40
TF252	Tam's Horror	1s	256MB	7 | 9 | 11 | 14 | 19	60
TF253	Definicraft Time	1s	256MB	8 | 11 | 14 | 17 | 25	75
TF254	King's Territory	1s	256MB	7 | 8 | 15 | 17 | 22 | 21	90
TF255	Robotic Cleaner	1s	256MB	13 | 20 | 12 | 14 | 17 | 29	105
TF256	Governor's Battle	1s	256MB	10 | 19 | 21 | 16 | 17 | 22 | 25	130
				Full Score	500

REMINDERS

- All problems are divided into subtasks. All test cases in a subtask must be passed to get points.
- You may attempt the problems using C++20 or Python 3 (secondary language). It is not guaranteed that the problems are solvable using secondary language.
- Unless otherwise specified, your output will be automatically fixed as follows:
  - Trailing spaces in each line will be removed.
  - An end-of-line character will be added to the end of the output if not present.
- 64-bit integers may be required in some problems, use `long long` in C++ if needed.
- All stories are hypothetically written. In case of coincidences, they are coincidences.

TF251 Standard Charter

Score	Limit	
40	1s	256MB

Time flies, it has already been a year since Daniel started his memorable and exciting university life. After successfully moving into the hostel, life has not ever been this happy with his roommates. They have become great friends and contact each other no matter if they are *in work and play, at school, at home, and always*.

"Urgh! Why is this game so difficult? My hands are having a cramp!" One of Daniel's roommates is playing a rhythm game one day in their hostel. Out of curiosity, Daniel wanted to see how this game works.

The rhythm game is played by tapping on tiles falling from four lanes when they land on the judgement line. Multiple tiles make a pattern and rhythm game charters would put different patterns in their charts.

Supposedly, to play a pattern, if the first tile of a pattern falls down from lane 1 or 2, left hand would be the starter hand. Otherwise, if it falls from lane 3 or 4, right hand would be the starter hand. Then, for the rest of the tiles, we would tap the tiles while alternating hands. For example, if we start with left hand, we tap the next tile with right hand, then left hand, and so on. According to Daniel's roommate, a pattern is considered "standard" if it does not involve any cross-hands to play, that is, **one hand does not cross another between two taps**.

Anyways, Daniel wants to know if the pattern is "standard", but human eyes are too slow sometimes to comprehend a BPM 9999 chart. Therefore, Daniel decides to ask you, a bright programming star, to write a program for him to tell if a pattern is "standard"!

INPUT

The first line consists of an integer N, representing the number of tiles in the pattern.

The next line consists of N integers, the i^{th} integer L_i represents the position of the i^{th} tile.

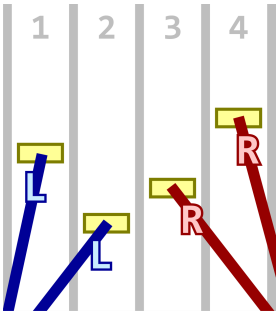
OUTPUT

If the pattern is not "standard", output `:(`.

Otherwise, output `:D`.

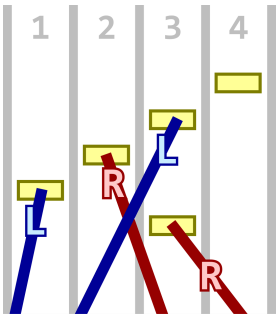
SAMPLE TESTS

	Input	Output
1	4 2 3 1 4	:D



	Input	Output
2	5 3 1 2 3 4	:(

Two hands must cross tapping the 3^{rd} tile and the 4^{th} tile.



CONSTRAINTS

$1 \leq N \leq 10^6$

$1 \leq L_i \leq 4$

SUBTASKS

	Points	Constraints
1	5	$N = 2$
2	9	$N = 3$
3	11	$1 \leq L_i \leq 2$
4	15	No additional constraints

TF252 Tam's Horror

Score	Limit
60	1s | 256MB

After playing the rhythm game for so long, Daniel's roommate felt quite hungry. This time, they decided to go out and find some hidden gems of restaurants near them, like this one rice noodles restaurant located in the back alley of a street, which coincidentally has the same name as the famous *Tam Jr. Mixian*! …except the menu looked a bit strange.

First of all, a type of noodles or vegetables would be chosen. The price for each of them are:

Type	*Price*
`Rice Noodles`	28
`Sweet Potato Noodles`	29
`Napa Cabbage`, `Lettuce`, `Bean Sprouts`	30

Then, a type of soup will be selected:

Soup	*Price*
`Clear Soup`	0
`Suan La Soup`	2
`Numbing Spicy Soup`, `San Suan Soup`	3
`Sha Sha Soup`	5

Then, the spiciness of the soup has to be decided also. Note that spicinesses are only applicable if `Numbing Spicy Soup`, `San Suan Soup` or `Sha Sha Soup` is selected.

Spiciness	*Price*
`0%`	40
`5%`	38
`10%`	36
⋮	⋮
`95%`	2
`100%`	0

Finally, N toppings would be decided. Note that toppings could be repeated and at least one topping should be ordered.

Toppings	*Price*
`Mushroom`	5
`Tofu Puffs`	5
`Fishball`	5
`Bean Curd Sheet`	5

Toppings	*Price*
`Pork`	6
`Beef`	6
`Bean Sprout`	6
`Beef Ball`	7
`Pork Belly`	7

"Why does 0% spiciness cost me $40??? The price is ridiculous!!! Where is the manager?????" Daniel's roommate complained. Little did he know, this is an unmanned restaurant and he could only order by using the kiosk beside them. However, Daniel noticed that the program in the kiosk is not working, so he decided to ask you, a bright programming star, to remotely write a new program for the kiosk while sitting in CAL!

INPUT

The first line consists of a string B, representing the type of rice noodles or vegetable ordered.

The second line consists of a string S, representing the type of soup ordered.

The third line consists of a string L, representing the spiciness of the soup. If spiciness is not applicable, $L =$ `N/A` instead.

The fourth line consists of N strings, each separated by `, ` (a comma followed by a space). The i^{th} string T_i representing the i^{th} toppings ordered.

OUTPUT

Output an integer, the total price required for the order.

SAMPLE TESTS

	Input	*Output*

1	`Rice Noodles`
	`Clear Soup`
	`N/A`
	`Pork, Beef Ball`
		`41`

Total price is $28 + 6 + 7 = 41$.

	Input	*Output*

2	`Bean Sprouts`
	`Sha Sha Soup`
	`0%`
	`Bean Sprout, Bean Sprout, Bean Sprout, Bean Sprout`
		`99`

Do not question.

3

Rice Noodles San Suan Soup 15% Pork Belly, Beef	78

4

Sweet Potato Noodles Suan La Soup N/A Mushroom	36

CONSTRAINTS

B, S and T_i must be one of the listed items in the problem statement

L must be either `N/A` or `[spiciness]%`, where `[spiciness]` is a multiple of 5

$1 \leq N \leq 30$

SUBTASKS

	Points	Constraints
1	7	$B =$ `Rice Noodles` $S =$ `Clear Soup` or `Suan La Soup` $N = 1$ $T_i =$ `Mushroom`, `Fishball`, `Pork` or `Beef`
2	9	$L =$ `100%` $N = 1$ $T_i =$ `Mushroom`, `Fishball`, `Pork` or `Beef`
3	11	$N = 1$ $T_i =$ `Mushroom`, `Fishball`, `Pork` or `Beef`
4	14	Price of every topping ordered $= 5$
5	19	No additional constraints

TF253 Definicraft Time

Score	Limit	
75	1s	256MB

After returning to the hostel, Daniel felt a bit bored, so he decided to launch and play his favourite game *Chemcraft*! However, he also felt bored crafting chemical compounds only! So, he decided to start a project and make a game that crafts everything besides chemical compounds. After intensive brainstorming, he ended up with the name *Definicraft*, which is *defini*tely not similar to *Chemcraft*.

The main operation of this game is "craft", which involves the putting of two items on a crafting table one by one. Generally, when you put an item called `[item1]` on the table, then another called `[item2]` on the table, the name of the new item would be `[item1][item2]`.

However, there are also M recipes, which determine a special name of an item when crafted with two specific items while ignoring the order of placements of the two items. In the j^{th} recipe, C_j is obtained when A_j and B_j are crafted together, regardless of the order of placements of A_j and B_j. For example, if one of the recipes states that `Heat` and `Ice` make `Water`, when you craft `Heat` and `Ice` (or `Ice` and `Heat`) together, the name of the new item is called `Water` instead of `HeatIce` (or `IceHeat`), as shown in the picture below:

Initially, there are N base items numbered from 1 to N, where item i has the name S_i. There is a list of items with K items and $K-1$ operations of crafting have to be done from the top of the list to the bottom. If more than one crafting operations have to be done, the rule is as follows:

- First, craft the top two items on the list into one item.
- Next, place the item crafted just now at the top of the list.
- Craft the top two items on the list and repeat until the final item is crafted.

As he is busy working on many game features, he decided to ask his friend Jack for help. However, as Jack is busy designing the interface of the game, he decided to leave this core mechanic for you, a bright programming star, to implement into the game!

INPUT

The first line consists of three integers N, M and K, representing the number of base items, the number of recipes and the number of items respectively.

The second line consists of N strings, each separated by a space. The i^{th} string S_i represents name of the item i.

The j^{th} line of the next M lines consists of three strings A_j, B_j and C_j, describing the j^{th} recipe where C_j is obtained when A_j and B_j is crafted together.

The final line consists of K integers, describing the list of items used.

OUTPUT

Output a string, the name of the resultant item.

SAMPLE TESTS

	Input	Output
1	2 0 2 meat ball 1 2	meatball

As there is no recipes for `meat` and `ball`, the name of the resultant item will be `meatball`.

2	5 3 2 are you smart maybe no are you smart smart maybe no no no yes 3 4	no
3	5 3 4 are you smart maybe no are you smart smart maybe no no no yes 1 2 4 5	yes

CONSTRAINTS

$1 \le N \le 10^5$

$0 \le M \le 10^5$

$2 \le K \le 10^5$

$1 \le \text{length of } S_i, A_j, B_j, C_j \le 20$

S_i, A_j, B_j, C_j consists of lowercase English letters (`a` - `z`) only

SUBTASKS

	Points	Constraints
1	8	$N = 2$ $M = 0$
2	11	$K = 2$ $M = 0$
3	14	$K = 2$
4	17	$2 \le K \le 100$
5	25	No additional constraints

TF254 King's Territory

Score	Limit
90	1s | 256MB

After developing the game for some time and gaining some progress, the team started to run out of ideas. Therefore, they decided to take inspiration from other games! Duncan suggested playing his favourite game *"Team Fight Territory (TFT)"*, and everyone agreed. So, as usual, Duncan started a stream of him playing this game and this time he decided to discover some new game modes in the game.

There is a system called "New Territories", and as the name suggests, players would fight for their land. Duncan wants to fight *King Charles IV* and *Queen Consort Nicoria* and get all of their land!

King Charles' kingdom is a long 1D land with length L, where locations are presented from 1 to L. There are N cities in the kingdom, where the i^{th} city is located at P_i. To improve governance, the kingdom is divided into K districts. The borders of the j^{th} district are l_j and r_j, where $l_j \leq r_j$ and $r_j < l_{j+1}$. The influence value of the j^{th} district is $(r_j - l_j + 1) \times \text{number of cities in it}$. The higher the influence value of a district is, the more difficult it is to be conquered. Wanting to protect all of his land, *King Charles* divides the kingdom in a way such that the minimum influence value among all districts is maximised.

Duncan wants to know the influence values of the districts so that they could prepare for a certain amount of fighting ability! Just as the team members are discussing the strategies, they shifted their focus on you, a bright programming star, to help with this plan of a mastermind! Could you write a program to maximise the minimum influence value among all districts?

INPUT

The first line consists of three integers N, L and K, representing the number of cities, the length of the kingdom and the number of districts respectively.

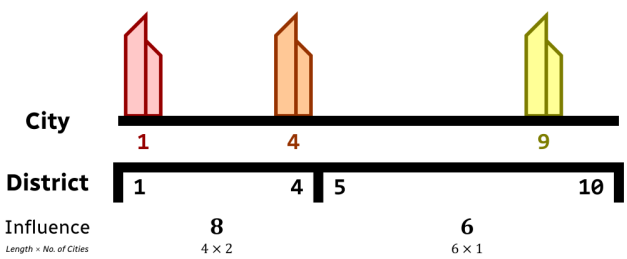
The second line consists of N integers, the i^{th} integer P_i represents the position of the i^{th} city.

OUTPUT

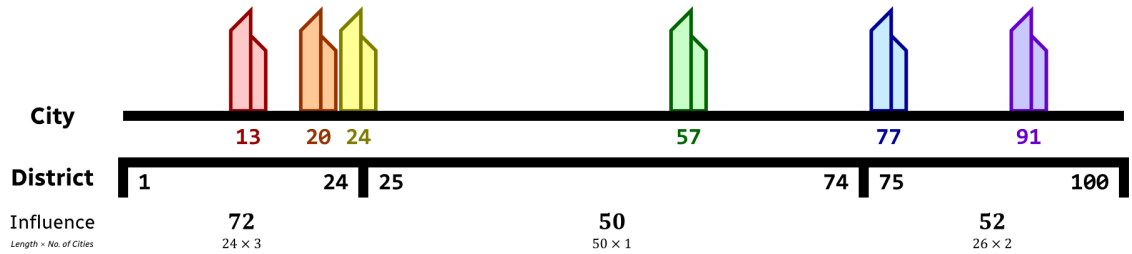
Output an integer, the maximum possible minimum influence value among all districts.

SAMPLE TESTS

	Input	Output
1	3 10 2 1 4 9	6



2	6 100 3 13 20 24 57 77 91	50



3	9 1000000000 5 1 10 100 1000 10000 100000 1000000 10000000 100000000 1000000000	499995

CONSTRAINTS

$2 \le K \le N \le \min(10^5, L)$

$2 \le L \le 10^9$

$1 \le P_1 < P_2 < \cdots < P_N \le L$

SUBTASKS

	Points	Constraints
1	7	$N = K = 2$ $1 \le L \le 10^6$
2	8	$N = K = 2$
3	15	$N = 3$ $1 \le L \le 10^6$
4	17	$N = K$ $1 \le L \le 10^6$
5	22	$1 \le L \le 10^6$
6	21	No additional constraints

TF255 Robotic Cleaner

Score	Limit	
105	1s	256MB

The team had so much fun playing *"Team Fight Territory (TFT)"* that they forgot about the project! But suddenly, Daniel heard some noise behind him…

Oh no! Daniel forgot to close the door of his hostel room and a robotic cleaner from another room accidentally came in! However, before returning the robot back to the owner, Daniel decided to close the door and let the robot clean the floor of his own room first! Now, Daniel wants to know how much the robot could clean up so that it could save him some time tidying up the room himself.

The room can be represented by a $N \times M$ grid where the top-left cell is $(1, 1)$ and the bottom-right cell is (N, M). There are K obstacles spreading in different cells, which the i^{th} obstacle is located at cell (R_i, C_i).

The robot can move to cells not occupied by obstacles. It has two operating modes:

Mode	Operations
Right	■ It will try to move from (r, c) to $(r, c + 1)$. ■ If it cannot move to $(r, c + 1)$, it will try to move to $(r + 1, c)$ instead and change to mode **left** immediately afterwards. ■ If it cannot move to both $(r, c + 1)$ and $(r + 1, c)$, it will terminate.
Left	■ It will try to move from (r, c) to $(r, c - 1)$. ■ If it cannot move to $(r, c - 1)$, it will try to move to $(r + 1, c)$ instead and change to mode **right** immediately afterwards. ■ If it cannot move to both $(r, c - 1)$ and $(r + 1, c)$, it will terminate.

The robot is initially located at $(1, 1)$ and it is in mode **right**. It will clean all cells it has ever visited (including $(1, 1)$ and the cell where the robot terminates).

As the team is still having fun playing *TFT*, Daniel decided to ask you, a bright programming star, to write a program to find the number of cells the robot can clean before it being terminated!

INPUT

The first line consists of three integers N, M and K, representing the number of rows, the number of columns and the number of obstacles respectively.

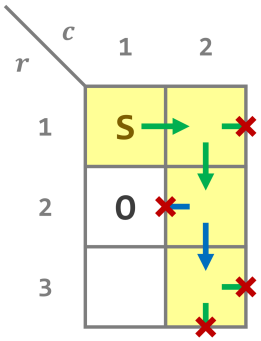
The i^{th} line of next K lines consists of two integers R_i and C_i, representing the coordinates of the i^{th} obstacle.

OUTPUT

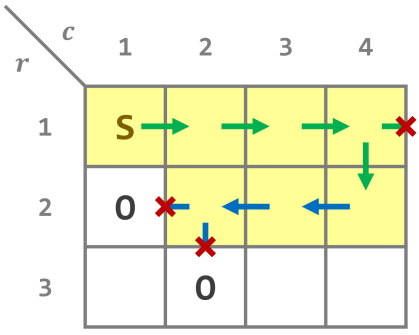
Output an integer, the number of cells cleaned by the robot before it being terminated.

SAMPLE TESTS

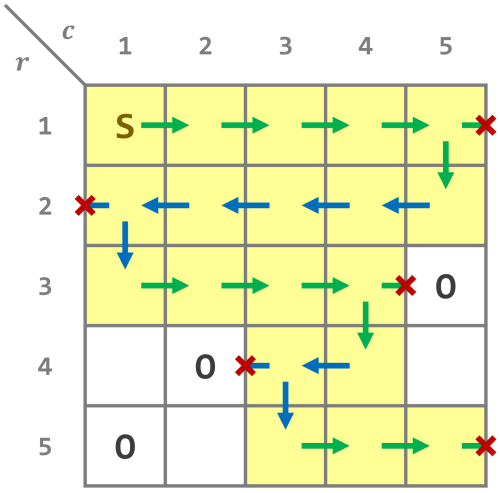
	Input	Output
1	3 2 1 2 1	4



	Input	Output
2	3 4 2 2 1 3 2	7



	Input	Output
3	5 5 3 3 5 4 2 5 1	19



CONSTRAINTS

$1 \le N, M \le 10^9$

$1 \le K \le \min(10^6, N \times M - 1)$

$1 \le R_i \le N$

$1 \le C_i \le M$

$(R_i, C_i) \ne (R_j, C_j) \ne (1, 1)$ for $i \ne j$

SUBTASKS

	Points	Constraints
1	13	$N = 3$ $M = 2$
2	20	$1 \le N, M \le 1000$
3	12	$K = 1$
4	14	$R_1 = R_2 = \cdots = R_K$
5	17	$K = 2$
6	29	No additional constraints

TF256 Governor's Battle

Score	Limit	
130	1s	256MB

Finally, a long day has ended. Daniel felt tired after watching the fierce fight over *King Charles'* land and quickly fell asleep… Suddenly, he found himself becoming a soldier in Duncan's army and standing on *King Charles'* land that they had just conquered! Duncan stood on the podium and announced, *"Before my great expedition to the United Kingdom, I am now going to appoint a Governor to rule this land!"* With that being said, he initiated a battle between soldiers in which the winner could become the Governor.

On the land, N soldiers are lining up in a straight line, indexed from 1 to N. The i^{th} soldier has a strength of S_i. To prove himself being the strongest Governor candidate, every soldier would only challenge other soldiers with strength higher than him. Commencing the battle, each of the soldiers' possible first target is the nearest soldier with strength higher than him either on his left (with index smaller than him) or on his right (with index greater than him).

"The position of Governor is mine!" Daniel is determined to win the battle, such that he can set up a school on this land called *The Winning Governor's Secondary School (TWGSS)*. However, lots of soldiers weaker than Daniel *(but somehow taller than him)* are blocking his sight! Therefore, Daniel sends you a message by telepathy, asking you, a bright programming star, to find the possible first target from both directions for each of the soldiers, so that he could start the battle with a great advantage.

INPUT

The first line consists of an integer N, representing the number of soldiers.

The second line consists of N integers, the i^{th} integer S_i represents the strength of the i^{th} soldier.

OUTPUT

Output N lines.

On the i^{th} line, output two integers, separated by a space, the index of the possible first target for the i^{th} soldier on his left side and right side respectively. If there is no possible target from a direction, output `-1` instead.

SAMPLE TESTS

	Input	Output
1	5 3 1 4 5 2	-1 3 1 3 -1 4 -1 -1 4 -1
2	7 10 20 30 40 30 20 10	-1 2 -1 3 -1 4 -1 -1 4 -1 5 -1 6 -1
3	10 1 4 4 2 8 16 4 8 1 2	-1 2 -1 5 -1 5 3 5 -1 6 -1 -1 6 8 6 -1 8 10 8 -1
4	5 2 2 3 3 2	-1 3 -1 3 -1 -1 -1 -1 4 -1

CONSTRAINTS

$1 \leq N \leq 10^6$

$1 \leq S_i \leq 10^9$

SUBTASKS

	Points	Constraints
1	10	$N = 3$
2	19	$1 \leq N \leq 8$
3	21	$1 \leq S_i \leq 2$
4	16	$1 \leq N \leq 5000$
5	17	$1 \leq S_i \leq 10$
6	22	$S_i \neq S_j$ for $i \neq j$
7	25	No additional constraints

HINTS

To optimise the runtime of your code, you can:

- Add `ios::sync_with_stdio(false),cin.tie(0),cout.tie(0);` to the beginning of the `main()` function.
- Use `'\n'` instead of `endl` for faster output speed.