Tom's Kitchen

1 sec / 10 sec **256 MB**

Tom's Kitchen is a very popular restaurant. One of the reasons for its popularity is that every single meal is prepared by at least K different chefs. Today, there are N meals to be prepared, with meal i needing A_i hours of work.

There are M chefs which Tom can hire to prepare all the meals but the chef j will work at most B_j hours. Additionally, even when he works less, he still wants to be paid for the full B_j hours. A chef can work on several meals for different amounts of time, but any meal will be properly prepared only if at least K chefs take part in preparing it and the total time they spend is exactly A_i. When a chef takes part in preparing a meal, he always works on it some positive integer number of hours.

Tom needs help in choosing the optimal subset of chefs such that the sum of hours where the chefs are getting paid without working is minimized.

Input. The first line contains the integers N, M, and K.

The second line contains N integers A_i and the third line M integers B_j.

Output. The only line should contain the number of hours the chefs spend not working but still getting paid when Tom chooses the optimal subset to hire. If there is no way to prepare all the N meals according to the rules described above, output "`Impossible`".

Example.

Input	Output

`1 2 2`
`5`
`3 4`
	`2`

Here Tom needs two chefs to work on the meal, so he must hire both that are available. Then it does not matter how they divide the work, as they end up working a total of 5 hours, but getting paid for $3 + 4 = 7$ hours, and thus getting paid for 2 extra hours.

Example.

Input	Output

`1 1 2`
`5`
`5`
	`Impossible`

Here Tom needs two chefs to work on the meal, but only one is available.

Example.

Input	Output

`3 3 3`
`3 3 2`
`3 3 3`
	`Impossible`

Here meal 3 can't be prepared by three chefs, as each would have to work for at least an hour, but the meal takes only 2 hours to prepare.

Grading. The test groups satisfy the following conditions:

1. (9 points) $1 \le N, K \le 300$, $1 \le M \le 2$, $1 \le A_i, B_j \le 300$.
2. (22 points) $1 \le N, K \le 300$, $1 \le M \le 15$, $1 \le A_i, B_j \le 300$.
3. (20 points) $1 \le N, M, A_i, B_j \le 300$, $K = 1$.
4. (21 points) $1 \le N, M, K, A_i, B_j \le 40$.
5. (28 points) $1 \le N, M, K, A_i, B_j \le 300$.

Necklace

1.5 sec / 15 sec **1 GB**

Jill and Jane are sisters. Last Christmas each of them got a string consisting of colorful beads. We can describe each color as a letter of the English alphabet ("`a`"… "`z`"), and each string of beads as a word.

The girls would like to create necklaces from their strings. They can turn each string into a necklace by removing some (possibly zero) beads from the ends, and then connecting the ends of the remaining part of the string. The resulting necklace can be rotated and turned over.

The sisters want their necklaces to look exactly the same, and also be as long as possible. What is the maximum length they could achieve?

Input. The first and the second line each contain a non-empty sequence consisting of no more than N lowercase characters, the decriptions of Jill's and Jane's strings respectively.

Output. The first line should contain a single positive integer: the maximum number of beads each girl's necklace can have in the end. It is guaranteed that a positive length can be achieved.

The second line should contain two integers: the starting positions of the necklaces in Jill's and Jane's string respectively. If there are several possiblities, output any one of them. The positions are numbered left to right starting from 0.

Example.

Input	Output
`zxyabcd`	`4`
`yxbadctz`	`3 2`

We can do as follows:
"`zxyabcd`" → "`---abcd`"
"`yxbadctz`" → "`--badc--`"
The strings "`abcd`" and "`badc`" result in identical necklaces.

Grading. In this task, your program receives full points for a test group if it correctly finds the longest possible necklaces in all test cases. If it finds in each test case necklaces at least half the length of the longest possible ones, it receives 20% of the points.

The test groups satisfy the following conditions:

1. (25 points) $N = 100$.
2. (20 points) $N = 400$.
3. (40 points) $N = 3000$.
4. (15 points) $N = 3000$.

The last group is a special case. It has the same time limit as above, but your solution is allowed to use only 3 MB of memory. Due to technical constraints, this sub-task is defined as a separate task (`necklace4`) on the contest server and you should submit your solution separately to `necklace1` and `necklace4`.

For C and C++ solutions, the 3 MB limit is applied directly. For Java and Python solutions, the memory limit enforced by the contest server is 3 MB above the memory requirements of the "Hello world" program. For Java, also the `-Xmx4224k -Xss256k -XX:MaxMetaspaceSize=8704k` command line options are passed to inform the JVM garbage collector of the limits.

Olympiads

1 sec / 10 sec **256 MB**

Two neighbouring cities send each year a team of K contestants to compete in K different events. Each contestant participates in all the events. The score of a team in an event is the highest score earned by any of that team's contestants in that event. The total score of a team is the sum of the scores of the team over all events. For example, if $K = 3$ and the contestants score $(4, 5, 3)$, $(7, 3, 6)$, and $(3, 4, 5)$ then the scores for the team in the events are $(7, 5, 6)$ and the total score of the team is 18.

Each city has a set of eligible contestants they can send to the competition. The cities have started arguing about not only which city has the best team, but also about which city has the better C-th best team for some integer C, where $C = 1$ corresponds to the best team, $C = 2$ is the second best team, and so on.

You are tasked with helping one of the cities finding out the expected score its C-th best team, considering all the different K-member teams they could compose from their eligible contestants. Two teams are considered different if they have at least one contestant different.

Input. The first line contains the integers N, K, and C, where N is the total number of eligible contestants in a city, K the size of the team ($K \leq N$), and C the index of the team we're interested in (C does not exceed the number of possible K-member teams).

Each of the following N lines contains K non-negative integers, the expected scores of one eligible contestant in the K events. No score will be greater than 10^6.

Output. The only line should contain the total score of the C-th best team.

Example. Input

```
5 4 4
7 0 4 9
3 0 8 4
1 1 3 7
5 1 3 4
4 2 2 9
```

Output

```
24
```

There are 5 possible teams and they would score 26, 26, 25, 24, and 22 points, so the 4-th best score is 24.

Grading. The test groups satisfy the following conditions:

1. (13 points) $1 \leq N \leq 500$, $1 \leq K \leq 2$, $1 \leq C \leq 2\,000$.
2. (31 points) $1 \leq N \leq 100$, $1 \leq K \leq 6$, $1 \leq C \leq 2\,000$.
3. (24 points) $1 \leq N \leq 500$, $1 \leq K \leq 6$, $1 \leq C \leq 2\,000$, no score is greater than 10.
4. (32 points) $1 \leq N \leq 500$, $1 \leq K \leq 6$, $1 \leq C \leq 2\,000$.