A Cities

There are n cities in Byteland, and k of them are important cities frequently visited by the king of Byteland.

There are also m roads in the country, each of them connecting two cities. Unfortunately, the condition of the roads is so bad that the king cannot drive through them at full speed with his sports car.

For each road, the cost for renovating it is known. Your task is to choose which roads will be renovated so that all k important cities are connected with renovated roads, and the total cost is as low as possible.

Input

The first input line contains three integers n, k and m: the number of cities, the number of important cities and the number of roads. The cities are numbered $1, 2, \dots, n$. The second input line contains k integers: the important cities.

Finally, the input contains m lines that describe the roads. Each line contains three integers a, b and c, meaning that there is a bidirectional road between cities a and b, and the renovation cost for the road is c.

You may assume that there is a route between any two cities.

Output

You should output the minimum total cost for renovating the roads so that the king can travel between all important cities with his sports car.

Example

Input:
```
4 3 6
1 3 4
1 2 4
1 3 9
1 4 6
2 3 2
2 4 5
3 4 8
```

Output:
```
11
```

Subtasks

In all subtasks $1 \leq c \leq 10^9$ and $n \geq k$.

Subtask 1 (22 points)

- $2 \leq k \leq 5$
- $n \leq 20$
- $1 \leq m \leq 40$

Subtask 2 (14 points)

- $2 \leq k \leq 3$
- $n \leq 10^5$
- $1 \leq m \leq 2 \cdot 10^5$

Subtask 3 (15 points)

- $2 \leq k \leq 4$
- $n \leq 1000$
- $1 \leq m \leq 2000$

Subtask 4 (23 points)

- $k = 4$
- $n \leq 10^5$
- $1 \leq m \leq 2 \cdot 10^5$

Subtask 5 (26 points)

- $k = 5$
- $n \leq 10^5$
- $1 \leq m \leq 2 \cdot 10^5$

B Maze

Uolevi has developed a game where the player collects coins in a maze.
At the moment, the problem is that the game is too easy. Could you design some challenging mazes for the game?

Each maze is a rectangular grid that consists of floors (`.`) and walls (`#`). One of the cells is a base (`x`), and some cells can contain coins (`o`). The player begins in the base, and can move left, right, up and down. The task of the player is to collect all coins in the maze and then return to the base.

The difficulty of a maze is the length of the shortest path that begins in the base, collects all coins and returns to the base.

Input

The input begins with an integer t: the number of mazes. After this, t lines follow. Each such line contains three integers n, m and k. This means that the size of the maze must be $n \times m$ cells and there must be exactly k coins.

Output

The output should contain t maze descriptions, separated by empty lines, in the same order as in the input. Each maze must be solvable.

Example

Input:
```
2
3 3 1
4 7 2
```

Output:
```
###
#.x
#o#

.o.####
.#..x.#
...##.#
###o...
```

The difficulty of the first maze is 4, and the difficulty of the second maze is 18.

Submission

This is an output only task and there is only one input file (maze.in). You can download the input file here. You have to submit an output file (maze.out) that contains all the mazes specified in the input file.

Grading

For each maze, your score is $\max(0, 100 - 3(d - x))$ where x is the difficulty of your maze and d is the difficulty of the most challenging maze found by the jury. Your total score for the task is the average of all scores rounded down to an integer.

C Swap

You are given a sequence of n numbers $x_1, x_2, \ldots, x_n$. Each number $1, 2, \ldots, n$ appears exactly once in the sequence.

You can modify the sequence using swaps. There are $n-1$ consecutive turns numbered $k = 2, 3, \ldots, n$. On turn k you can either swap values x_k and $x_{\lfloor k/2 \rfloor}$ in the sequence or do nothing.

Sequence $a_1, a_2, \ldots, a_n$ is lexicographically smaller than sequence $b_1, b_2, \ldots, b_n$ if there exists an index j ($1 \le j \le n$) such that $a_k = b_k$ for all $k < j$ and $a_j < b_j$.

What is the lexicographically minimal sequence you can obtain?

Input

The first input line contains an integer n.

The second input line contains n integers: the numbers in the sequence.

Output

You should output n integers: the lexicographically minimal sequence.

Example

Input:
```
5
3 4 2 5 1
```

Output:
```
2 1 3 4 5
```

Subtask 1 (10 points)

- $1 \le n \le 20$

Subtask 2 (11 points)

- $1 \le n \le 40$

Subtask 3 (27 points)

- $1 \le n \le 1000$

Subtask 4 (20 points)

- $1 \le n \le 5 \cdot 10^4$

Subtask 5 (32 points)

- $1 \le n \le 2 \cdot 10^5$