



| task | type | time limit | memory limit |
|----------|-------------|------------|--------------|
| A Cities | standard | 2.00 s | 256 MB |
| B Maze | output only | N/A | N/A |
| C Swap | standard | 1.00 s | 256 MB |

A Cities

In Byteland gibt es n Städte, und k davon sind wichtige Städte, die häufig vom König von Byteland besucht werden.

Es gibt außerdem m Straßen in Byteland, von denen jede zwei Städte verbindet. Leider ist der Zustand der Straßen so schlecht, dass der König sie nicht mit seinem Sportwagen mit maximaler Geschwindigkeit befahren kann.

Für jede Straße sind die Kosten um sie zu renovieren bekannt. Deine Aufgabe ist es, auszuwählen, welche Straßen renoviert werden, so dass alle k wichtigen Städte durch renovierte Straßen verbunden sind, und die Gesamtkosten so gering wie möglich sind.

Eingabe

Die erste Zeile der Eingabe enthält drei Integer n , k und m : die Anzahl der Städte, die Anzahl der wichtigen Städte und die Anzahl der Straßen. Die Städte sind von 1 bis n nummeriert. Die zweite Zeile der Eingabe enthält k Integer: die wichtigen Städte.

Die letzten m Zeilen der Eingabe beschreiben die Straßen. Jede Zeile enthält drei Integer a , b und c . Dies bedeutet, dass eine in beide Richtungen befahrbare Straße zwischen a und b verläuft, und die Renovationskosten für diese Straße sind c .

Du kannst davon ausgehen, dass zwischen zwei beliebigen Städten jeweils eine Route existiert.

Ausgabe

Gebe die minimalen Gesamtkosten für die Renovierung der Straßen an, so dass der König zwischen allen wichtigen Städten mit seinem Sportwagen fahren kann.

Beispiel

Eingabe:

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

Ausgabe:

```
11
```

Teilaufgaben

In allen Teilaufgaben ist $1 \leq c \leq 10^9$ und $n \geq k$.

Teilaufgabe 1 (22 Punkte)

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

Teilaufgabe 2 (14 Punkte)

- $2 \leq k \leq 3$
- $n \leq 10^5$

- $1 \leq m \leq 2 \cdot 10^5$

Teilaufgabe 3 (15 Punkte)

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

Teilaufgabe 4 (23 Punkte)

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

Teilaufgabe 5 (26 Punkte)

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

B Maze

Uolevi hat ein Spiel entwickelt, bei dem die Spieler Münzen in einem Labyrinth sammeln. Aktuell gibt es das Problem, dass das Spiel zu einfach ist. Kannst du anspruchsvolle Labyrinth für das Spiel erstellen?

Jedes Labyrinth ist ein rechteckiges Raster, dass aus Böden (.) und Wänden (#) besteht. Eins der Felder ist ein Startpunkt (x), und manche Felder enthalten Münzen (o). Der Spieler startet am Startpunkt, und kann sich nach links, rechts, oben und unten bewegen. Die Aufgabe des Spielers ist es, alle Münzen in dem Labyrinth einzusammeln und dann zum Startpunkt zurückzukehren.

Die Schwierigkeit eines Labyrinths ist die Länge des kürzesten Pfades, der am Startpunkt anfängt, alle Münzen einsammelt und zum Startpunkt zurückkehrt.

Eingabe

Die Eingabe beginnt mit einem Integer t : die Anzahl der Labyrinth. Danach folgen t Zeilen. Jede dieser Zeilen enthält drei Integer n , m und k . Das bedeutet, dass die Größe des Labyrinths $n \times m$ Felder sein und dass es genau k Münzen geben muss.

Output

Die Ausgabe soll t Labyrinthbeschreibungen enthalten, getrennt durch Leerzeilen, in der gleichen Reihenfolge wie in der Eingabe. Jedes Labyrinth muss lösbar sein.

Beispiel

Eingabe:

```
2
3 3 1
4 7 2
```

Ausgabe:

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

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

Die Schwierigkeit des ersten Labyrinths ist 4, und die Schwierigkeit des zweiten Labyrinths ist 18.

Einsendung

Dies ist eine output only Aufgabe und es gibt nur eine Eingabedatei (maze.in). Du kannst die Eingabedatei [hier](#) herunterladen. Du musst eine Ausgabedatei (maze.out) einsenden, die alle Labyrinth enthält, die in der Eingabedatei spezifiziert sind.

Bewertung

Für jedes Labyrinth ist deine Punktzahl $\max(0, 100 - 3(d - x))$, wobei x die Schwierigkeit deines Labyrinths und d die Schwierigkeit des schwierigsten Labyrinths ist, das die Jury gefunden hat. Deine Gesamtpunktzahl für die Aufgabe ist der Durchschnitt aller Punktzahlen abgerundet auf einen Integer.

C Swap

Gegeben ist eine Folge von n Zahlen x_1, x_2, \dots, x_n . Jede Zahl $1, 2, \dots, n$ tritt genau einmal in der Folge auf.

Du kannst die Folge mit Hilfe von Vertauschungen verändern. Es gibt $n - 1$ aufeinanderfolgende Züge, nummeriert $k = 2, 3, \dots, n$. Im Zug k kannst du entweder die Werte x_k und $x_{\lfloor k/2 \rfloor}$ in der Folge vertauschen oder nichts tun.

Eine Folge a_1, a_2, \dots, a_n ist lexikographisch kleiner als eine Folge b_1, b_2, \dots, b_n , falls es einen Index j ($1 \leq j \leq n$) gibt, so dass $a_k = b_k$ für alle $k < j$ und $a_j < b_j$.

Was ist die lexikographisch kleinste Folge, die du erzeugen kannst?

Eingabe

Die erste Zeile der Eingabe enthält einen Integer n .

Die zweite Zeile der Eingabe enthält n Integer: die Zahlen der Folge.

Ausgabe

Gib n Integer aus: die lexikographisch kleinste Folge.

Beispiel

Eingabe:

5
3 4 2 5 1

Ausgabe:

2 1 3 4 5

Teilaufgabe 1 (10 Punkte)

- $1 \leq n \leq 20$

Teilaufgabe 2 (11 Punkte)

- $1 \leq n \leq 40$

Teilaufgabe 3 (27 Punkte)

- $1 \leq n \leq 1000$

Teilaufgabe 4 (20 Punkte)

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

Teilaufgabe 5 (32 Punkte)

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