A - ABC400 Party

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 100 points

Problem Statement

In the ceremony commemorating ABC400, we want to arrange 400 people in a rectangular formation of A rows and B columns without any gaps.

You are given a positive integer A. Print the value of a positive integer B for which such an arrangement is possible. If there is no such positive integer B, print -1.

Constraints

- A is an integer between 1 and 400, inclusive.

Input

The input is given from Standard Input in the following format:

$$A$$

Output

Print the value of B or -1 as specified by the problem statement.

Sample Input 1

```
10
```

Sample Output 1

```
40
```

We can arrange 400 people in 10 rows and 40 columns.

Sample Input 2

```
11
```

Sample Output 2

```
-1
```

Sample Input 3

```
400
```

Sample Output 3

```
1
```

B - Sum of Geometric Series

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 200 points

Problem Statement

You are given two positive integers N and M.

Let $X = \sum_{i=0}^{M} N^i$. If $X \leq 10^9$, print the value of X. If $X > 10^9$, print `inf`.

Constraints

- $1 \leq N \leq 10^9$
- $1 \leq M \leq 100$
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$N \quad M$$

Output

Print the value of X or `inf` as specified by the problem statement.

Sample Input 1

```
7 3
```

Sample Output 1

```
400
```

$X = 1 + 7 + 49 + 343 = 400$. Since $400 \leq 10^9$, print 400.

Sample Input 2

```
1000000 2
```

Sample Output 2

```
inf
```

$X = 1000001000001 > 10^9$, so print `inf`.

Sample Input 3

```
999999999 1
```

Sample Output 3

```
1000000000
```

Sample Input 4

```
998244353 99
```

Sample Output 4

```
inf
```

C - 2^a b^2

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 350 points

Problem Statement

A positive integer X is called a good integer if and only if it satisfies the following condition:

- There exists a pair of positive integers (a, b) such that $X = 2^a \times b^2$.

For example, 400 is a good integer because $400 = 2^2 \times 10^2$.

Given a positive integer N, find the number of good integers between 1 and N, inclusive.

Constraints

- $1 \leq N \leq 10^{18}$
- N is an integer.

Input

The input is given from Standard Input in the following format:

$$N$$

Output

Print the number of good integers between 1 and N, inclusive.

Sample Input 1

```
20
```

Sample Output 1

```
5
```

There are five good integers between 1 and 20: 2, 4, 8, 16, and 18.
Thus, print 5.

Sample Input 2

```
400
```

Sample Output 2

```
24
```

Sample Input 3

```
1234567890
```

Sample Output 3

```
42413
```

Note that the input might not fit in a 32-bit integer type.

D - Takahashi the Wall Breaker

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 400 points

Problem Statement

Takahashi is about to go buy eel at a fish shop.

The town where he lives is divided into a grid of H rows and W columns. Each cell is either a road or a wall. Let us denote the cell at the i-th row from the top $(1 \leq i \leq H)$ and the j-th column from the left $(1 \leq j \leq W)$ as cell (i, j).
Information about each cell is given by H strings $S_1, S_2, \ldots, S_H$, each of length W. Specifically, if the j-th character of S_i $(1 \leq i \leq H, 1 \leq j \leq W)$ is `.`, cell (i, j) is a road; if it is `#`, cell (i, j) is a wall.

He can repeatedly perform the following two types of actions in any order:

- Move to an adjacent cell (up, down, left, or right) that is within the town and is a road.
- Choose one of the four directions (up, down, left, or right) and perform a **front kick** in that direction.
  When he performs a front kick, for each of the cells at most 2 steps away in that direction from the cell he is currently in, if that cell is a wall, it becomes a road.
  If some of the cells at most 2 steps away are outside the town, a front kick can still be performed, but anything outside the town does not change.

He starts in cell (A, B), and he wants to move to the fish shop in cell (C, D).
It is guaranteed that both the cell where he starts and the cell with the fish shop are roads.
Find the minimum **number of front kicks** he needs in order to reach the fish shop.

Constraints

- $1 \leq H \leq 1000$
- $1 \leq W \leq 1000$
- Each S_i is a string of length W consisting of `.` and `#`.
- $1 \leq A, C \leq H$
- $1 \leq B, D \leq W$
- $(A, B) \neq (C, D)$
- H, W, A, B, C, and D are integers.
- The cell where Takahashi starts and the cell with the fish shop are roads.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
H \quad W \\
S_1 \\
S_2 \\
\vdots \\
S_H \\
A \quad B \quad C \quad D
\end{array}
$$

Output

Print the minimum number of front kicks needed for Takahashi to reach the fish shop.

Sample Input 1

```
10 10
..........
#########.
#.......#.
#..####.#.
##....#.#.
#####.#.#.
.##.#.#.#.
###.#.#.#.
###.#.#.#.
#.....#...
1 1 7 1
```

Sample Output 1

```
1
```

Takahashi starts in cell $(1, 1)$.
By repeatedly moving to adjacent road cells, he can reach cell $(7, 4)$.
If he performs a front kick to the left from cell $(7, 4)$, cells $(7, 3)$ and $(7, 2)$ turn from walls to roads.
Then, by continuing to move through road cells (including those that have become roads), he can reach the fish shop in cell $(7, 1)$.

In this case, the number of front kicks performed is 1, and it is impossible to reach the fish shop without performing any front kicks, so print 1.

Sample Input 2

```
2 2
.#
#.
1 1 2 2
```

Sample Output 2

```
1
```

Takahashi starts in cell $(1, 1)$.
When he performs a front kick to the right, cell $(1, 2)$ turns from a wall to a road.
The cell two steps to the right of $(1, 1)$ is outside the town, so it does not change.
Then, he can move to cell $(1, 2)$ and then to the fish shop in cell $(2, 2)$.

In this case, the number of front kicks performed is 1, and it is impossible to reach the fish shop without performing any front kicks, so print 1.

Sample Input 3

```
1 3
.#.
1 1 1 3
```

Sample Output 3

```
1
```

When performing a front kick, it is fine if the fish shop's cell is within the cells that could be turned into a road. Specifically, the fish shop's cell is a road from the beginning, so it remains unchanged; particularly, the shop is not destroyed by the front kick.

Sample Input 4

```
20 20
####################
##...##....###...###
#.....#.....#.....##
#..#..#..#..#..#..##
#..#..#....##..#####
#.....#.....#..#####
#.....#..#..#..#..##
#..#..#.....#.....##
#..#..#....###...###
####################
####################
##..#..##...###...##
##..#..#.....#.....#
##..#..#..#..#..#..#
##..#..#..#..#..#..#
##.....#..#..#..#..#
###....#..#..#..#..#
#####..#.....#.....#
#####..##...###...##
####################
3 3 18 18
```

Sample Output 4

```
3
```

E - Ringo's Favorite Numbers 3

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 425 points

Problem Statement

A positive integer N is a **400 number** if and only if it satisfies both of the following two conditions:

- N has exactly 2 distinct prime factors.
- For each prime factor p of N, p divides N an even number of times. More formally, the maximum non-negative integer k such that p^k divides N is even.

Process Q queries. Each query gives you an integer A, so find the largest 400 number not exceeding A. Under the constraints of this problem, a 400 number not exceeding A always exists.

Constraints

- $1 \leq Q \leq 2 \times 10^5$
- For each query, $36 \leq A \leq 10^{12}$.
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{aligned}
&Q \\
&\mathrm{query}_1 \\
&\mathrm{query}_2 \\
&\vdots \\
&\mathrm{query}_Q
\end{aligned}
$$

Here, query_i is the i-th query, given in the following format:

$$A$$

Output

Print Q lines. The i-th line should contain the answer to the i-th query.

Sample Input 1

```
5
404
36
60
1000000000000
123456789
```

Sample Output 1

```
400
36
36
1000000000000
123454321
```

Let us explain the first query.

There are exactly 2 prime factors of 400: 2 and 5. Also, 2 divides 400 four times and 5 divides it twice, so 400 is a 400 number. None of 401, 402, 403, and 404 is a 400 number, so the answer is 400.

F - Happy Birthday! 3

Time Limit: 3 sec / Memory Limit: 1024 MiB

Score : 550 points

Problem Statement

There is a circular cake that has been cut into N equal slices by its radii.

Each piece is labeled with an integer from 1 to N in clockwise order, and for each integer i with $1 \leq i \leq N$, the piece i is also referred to as piece $N + i$.

Initially, every piece's color is color 0.

You can perform the following operation any number of times:

- Choose integers a, b, and c such that $1 \leq a, b, c \leq N$. For each integer i with $0 \leq i < b$, change the color of piece $a + i$ to color c. The cost of this operation is $b + X_c$.

You want each piece i (for $1 \leq i \leq N$) to have color C_i. Find the minimum total cost of operations needed to achieve this.

Constraints

- $1 \leq N \leq 400$
- $1 \leq C_i \leq N$
- $1 \leq X_i \leq 10^9$
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \\
C_1 \quad C_2 \quad \dots \quad C_N \\
X_1 \quad X_2 \quad \dots \quad X_N
\end{array}
$$

Output

Print the answer.

Sample Input 1

```
6
1 4 2 1 2 5
1 2 3 4 5 6
```

Sample Output 1

```
20
```

Let A_i denote the color of piece i. Initially, $(A_1, A_2, A_3, A_4, A_5, A_6) = (0, 0, 0, 0, 0, 0)$.

Performing an operation with $(a, b, c) = (2, 1, 4)$ changes $(A_1, A_2, A_3, A_4, A_5, A_6)$ to $(0, 4, 0, 0, 0, 0)$.

Performing an operation with $(a, b, c) = (3, 3, 2)$ changes $(A_1, A_2, A_3, A_4, A_5, A_6)$ to $(0, 4, 2, 2, 2, 0)$.

Performing an operation with $(a, b, c) = (1, 1, 1)$ changes $(A_1, A_2, A_3, A_4, A_5, A_6)$ to $(1, 4, 2, 2, 2, 0)$.

Performing an operation with $(a, b, c) = (4, 1, 1)$ changes $(A_1, A_2, A_3, A_4, A_5, A_6)$ to $(1, 4, 2, 1, 2, 0)$.

Performing an operation with $(a, b, c) = (6, 1, 5)$ changes $(A_1, A_2, A_3, A_4, A_5, A_6)$ to $(1, 4, 2, 1, 2, 5)$.

In this case, the total cost is $5 + 5 + 2 + 2 + 6 = 20$.

Sample Input 2

```
5
1 2 3 4 5
1000000000 1000000000 1000000000 1000000000 1000000000
```

Sample Output 2

```
5000000005
```

Sample Input 3

```
8
2 3 3 1 2 1 3 1
3 4 1 2 5 3 1 2
```

Sample Output 3

```
23
```

G - Patisserie ABC 3

Time Limit: 3 sec / Memory Limit: 1024 MiB

Score : 625 points

Problem Statement

Takahashi, a patissier working at the ABC pastry shop, decided to sell assorted cakes to commemorate AtCoder Beginner Contest 400.

The shop sells N kinds of cakes: cake 1, cake 2, $\ldots$, cake N.
Each cake has three non-negative integer values: beauty, tastiness, and popularity. Specifically, cake i has beauty X_i, tastiness Y_i, and popularity Z_i.

He considers pairing up these cakes into K pairs without overlaps.
Formally, he will choose $2K$ **distinct** integers $a_1, b_1, a_2, b_2, \ldots, a_K, b_K$ between 1 and N (inclusive), and pair cake a_i with cake b_i.
The price of a pair formed by cakes a_i and b_i is $\max(X_{a_i} + X_{b_i},\ Y_{a_i} + Y_{b_i},\ Z_{a_i} + Z_{b_i})$.
Here, $\max(P, Q, R)$ denotes the greatest value among P, Q, R.

Find the maximum possible total price of the K pairs.

You are given T test cases; solve each of them.

Constraints

- $1 \leq T \leq 1000$
- $2 \leq N \leq 10^5$
- The sum of N over all test cases in each input file is at most 10^5.
- $1 \leq K \leq \lfloor \frac{N}{2} \rfloor$ (For a real number x, $\lfloor x \rfloor$ denotes the greatest integer not exceeding x.)
- $0 \leq X_i, Y_i, Z_i \leq 10^9$
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
T \\
\mathrm{case}_1 \\
\mathrm{case}_2 \\
\vdots \\
\mathrm{case}_T
\end{array}
$$

case_i represents the i-th test case. Each test case is given in the following format:

$$
\begin{array}{l}
N \quad K \\
X_1 \quad Y_1 \quad Z_1 \\
X_2 \quad Y_2 \quad Z_2 \\
\vdots \\
X_N \quad Y_N \quad Z_N
\end{array}
$$

Output

Print T lines. The i-th line $(1 \leq i \leq T)$ should contain the answer to the i-th test case.

Sample Input 1

```
1
3 1
6 3 8
3 5 0
2 7 3
```

Sample Output 1

```
12
```

We form one pair out of three cakes.

If we pair cake 1 with cake 2, the price is $\max(6+3,\ 3+5,\ 8+0) = 9$.
If we pair cake 1 with cake 3, the price is $\max(6+2,\ 3+7,\ 8+3) = 11$.
If we pair cake 2 with cake 3, the price is $\max(3+2,\ 5+7,\ 0+3) = 12$.

Hence, pairing cake 2 with cake 3 gives the highest price, which is 12.

Sample Input 2

```
2
5 2
1 2 3
1 2 3
1 2 3
1 2 3
100 100 200
6 2
21 74 25
44 71 80
46 28 96
1 74 24
81 83 16
55 31 1
```

Sample Output 2

```
209
333
```

Note that each cake can appear in at most one pair.
Also note that there can be different cakes with identical values of beauty, tastiness, and popularity.

For the first test case, pairing cake 1 with cake 2 gives a price of 6, pairing cake 3 with cake 5 gives a price of 203, and choosing these two pairs yields a total price of 209, which is the maximum.

For the second test case, pairing cake 2 with cake 3 gives a price of 176, pairing cake 4 with cake 5 gives a price of 157, and choosing these two pairs yields a total price of 333, which is the maximum.