PROGRAMMING ASSIGNMENT 2

DUE: Wednesday, March 23, 11:59 PM. **No late submissions allowed.**
DO NOT COPY. ACKNOWLEDGE YOUR SOURCES.

Please read `http://www.student.cs.uwaterloo.ca/~cs341` for general instructions and policies.

Problem Description

Problem Statement

For this assignment, you are asked to implement an algorithm to solve Problem 2 of Assignment 6. That is, given a directed weighted graph G and a value h, you task is to compute, for all pairs i, j, the minimum weight of walks from i to j that use h edges. Note that edge weights can be negative.

Input Format

The first line of input consists of three integers: n, the number of vertices in the graph, m, the number of edges in the graph, and h, the required number of edges in the walks. The vertices of the graph are numbered 1 through n.
m lines of input follow. The i-th line consists of three integers u_i, v_i, w_i, $(1 \leq u_i, v_i \leq n)$, representing a **directed** edge from u_i to v_i, with weight w_i.
It is guaranteed that the graph has no self-loops or parallel edges. That is, $u_i \neq v_i$ for all $1 \leq i \leq m$, and $(u_i, v_i) \neq (u_j, v_j)$ for all $i \neq j$.

Output Format

Output n lines, each line with n entries. On the i-th line, the j-th entry should either be the minimum weight of walks from node i to node j that use h edges, or the character `x` (lowercase x) if no such walk exists. Separate two adjacent entries on the same line by a space.

Constraints

For all test cases, $1 \leq n \leq 80, 0 \leq m \leq n(n-1), 0 \leq h \leq 10^9, -10^9 \leq w_i \leq 10^9$.
For test cases worth 80% of the points, $0 \leq h \leq 80$.

Hint: The answers may exceed 32-bit integer range.

Sample Input 1

```
2 2 5
1 2 -3
2 1 2
```

Sample Output 1

```
x -5
0 x
```

Sample Input 2

```
4 5 5
1 2 -3
2 1 2
1 3 10
3 4 10
4 1 10
```

Sample Output 2

```
29 -5 8 50
0 29 42 21
50 16 29 x
8 37 50 29
```

Sample Input 3

```
3 2 1
3 2 -100
2 1 -100
```

Sample Output 3

```
x x x
-100 x x
x -100 x
```

Submission Instructions

- Submit your solution on Marmoset.
- You can choose to code in either C++ or Python.
- Name your program `prog2.cpp`/`prog2.py`.
- **Time limit**: 2 seconds (C++) / ~~6~~ 14 seconds (Python) for each test case.
- Compilation command for C++: `g++ -std=c++14 prog2.cpp -O3 -o prog2`.
- Execution command for Python: `python3 prog2.py`.
- Read from standard input and write to standard output.

- There will be several test cases, worth a total of 40 points. The public tests are worth 10 points and the secret tests are worth 30 points. The public tests (input only, not the answer) will be made available under a separate file.
- We will take the submission with the highest score. Please, however, refrain from excessive submissions.
- General collaboration policy applies. Please acknowledge your collaborator(s) by adding a comment in the beginning of your code.
- FAQ and updates will be posted on Piazza when necessary.