A Floyd-Warshall Problem

Time limit: 4.5s **Memory limit:** 1G

Here is an incorrect implementation of Floyd-Warshall.

```
floyd_warshall(dist, n):
  # Assume dist[i][j] is positive infinity if there is no edge between them
  for i ranging from 1 to n:
    for j ranging from 1 to n:
      for k ranging from 1 to n:
        dist[i][j] = min(dist[i][j], dist[i][k] + dist[k][j])
```

Here is an attempt at patching it.

```
floyd_warshall_patch1(dist, n, k):
  # dist[i][i] is zero
  # dist[i][j] is otherwise the weighted of the directed edge from i to j if it exists
  # dist[i][j] is otherwise positive infinity
  for i ranging from 1 to k:
    floyd_warshall(dist, n)
```

Here is another attempt at patching it.

```
floyd_warshall_patch2(dist, n)
  # dist[i][i] is zero
  # dist[i][j] is otherwise the weighted of the directed edge from i to j if it exists
  # dist[i][j] is otherwise positive infinity
  for i ranging from 1 to n:
    for j ranging from 1 to n:
      for k ranging from 1 to n:
        dist[j][k] = min(dist[j][k], dist[j][i] + dist[i][k])
```

Your job is to construct a directed graph with N vertices and M edges such that, given a parameter K, the output of `floyd_warshall_patch1` when given K matches the output of `floyd_warshall_patch2` on the given graph, but the output of `floyd_warshall_patch1` when given $K - 1$ does not.

Input Specification

The first line contains three space-separated integers, N, M, and K.

You may assume $2 \leq N \leq 100$, $N - 1 \leq M \leq N^2 - N$, and $1 \leq K \leq 3$.

Output Specification

If no such directed graph exists, output `NO` on a single line.

Otherwise, output $M + 1$ lines. The first line must be `YES`.

Each line after that should contain three space-separated positive integers. The first two, A and B, should indicate the presence of a directed edge going from A to B. Only one such edge may exist in your graph. Furthermore, $A \neq B$ and $1 \leq A, B \leq N$. The third integer is the weight of your edge. The weight must be a positive integer not exceeding 10^9.

Note: Depending on how easy this original task is, additional constraints may be added.

Sample Input 1

```
2 1 1
```

Sample Output 1

```
NO
```