7460 Maximum Cut Order

Max uses a tree $G(V, E)$ with undirected edges to model a network topology. Suppose that the vertex set $V = \{1, \ldots, n\}$ and each undirected edge $\{i, j\} \in E$ has an integral capacity $c(\{i, j\}) \geq 0$. For this problem, Max likes a specific tree, where the edge set is $E = \{\{i, \lfloor i/2 \rfloor\} : i = 2, \ldots, n\}$, that is for each vertex $i \in V$ with $i > 1$ there is an edge $\{i, \lfloor i/2 \rfloor\}$. For edge $e = \{i, j\} \in E$, the capacity of e is $c(e) = |i - j| \ \% \ m$, for some given positive integer m as the modulus, i.e., $c(e)$ is the remainder after $|i - j|$ is divided by m.

To maximize the communication throughput, Max wants to order the vertices in the sequence $v_1, \ldots, v_n$, which is a permutation of V, such that for all $i \in \{2, \ldots, n\}$ the *cut capacity* between v_i and $\{v_1, \ldots, v_{i-1}\}$ is maximized over all possible $v_j \in V - \{v_1, \ldots, v_{i-1}\}$. Max calls such order a *Maximum Cut Order*. The cut capacity between two disjoint sets $A, B \subseteq V$ is the sum of capacities of edges with one end vertex in A and the other in B. For convenience, define the edge subset between A and B as $E(A, B) = \{e : e = \{i, j\} \in E, i \in A, j \in B\}$. A *Maximum Cut Order* $v_1, v_2, \ldots, v_n$ satisfies that for all $i \in \{2, \ldots, n\}$:

$$\sum_{e \in E(\{v_1, \ldots, v_{i-1}\}, \{v_i\})} c(e) = \max_{j \in \{i, \ldots, n\}} \sum_{e \in E(\{v_1, \ldots, v_{i-1}\}, \{v_j\})} c(e)$$

If there is a tie when determining v_i, we choose the vertex with smaller index, i.e., following the lexicographical order.

For example, as in the above graph with $n = 7$, we have:

- $V = \{1, 2, 3, 4, 5, 6, 7\}$ and
- $E = \{\{1, 2\}, \{1, 3\}, \{2, 4\}, \{2, 5\}, \{3, 6\}, \{3, 7\}\}$.

Note that $c(\{i, j\}) = |i - j| \ \% \ 2$. Let the starting vertex v_1 be 1. Then the Maximum Cut Order is: 1, 2, 5, 3, 6, 4, 7. If $v_1 = 7$, then the order is: 7, 3, 6, 1, 2, 5, 4.

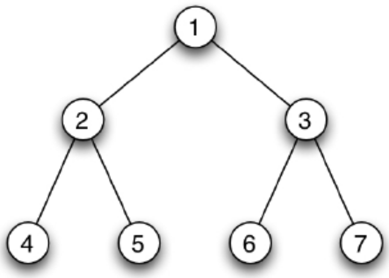

Figure 2: Three timed event flows.

Your task is to write a program to find the Maximum Cut Order of $G(V, E)$ from a given starting vertex.

Technical Specification

1. k: the number of test case, $k \leq 10$.
2. n: the number of vertices, $2 \leq n \leq 500000$.
3. s: the starting vertex, $s \leq n$.
4. m: the capacity modulus, $2 \leq m \leq 10000$.

Input

The first line of the input file contains an integer $k (\leq 10)$ indicating the number of test cases. Each test case consists of three integers n, s and m, separated with space(s), in a line, where $n (\leq 500000)$ indicates the number of vertices, $s (\leq n)$ indicates the starting vertex, and $m (\leq 10000)$ is the modulus for capacity. The capacity for an edge $e = \{i, j\}$ is simply $|i - j| \ \% \ m$.

Output

For each test case, output the Maximum Cut Order in a line, where the first element is the starting vertex. Separate adjacent vertices with a space.

Sample Input

```
3
2 2 5
7 1 2
7 7 2
```

Sample Output

```
2 1
1 2 5 3 6 4 7
7 3 6 1 2 5 4
```