Programming Part

Graph (20 pts)

Problem Description

In this problem, we ask you to implement the standard depth first search (DFS) algorithm to traverse a given directed graph $G = \{V, E\}$. Then, according to the results, classify each edge in E to be a tree edge, a back edge, a forward edge, or a cross edge.

The pseudo code of the DFS algorithm is given below, which is identical to the one in Chapter 20.3 of the textbook.

```
DFS(G)
1  for each vertex u ∈ G.V
2      u.color = WHITE
3      u.π = NIL
4  time = 0
5  for each vertex u ∈ G.V
6      if u.color == WHITE
7          DFS-VISIT(G, u)
```

```
DFS-VISIT(G, u)
 1  time = time + 1
 2  u.d = time
 3  u.color = GRAY
 4  for each v ∈ G.Adj[u]
 5      if v.color == WHITE
 6          v.π = u
 7          DFS-VISIT(G,v)
 8  u.color = BLACK
 9  time = time + 1
10  u.f = time
```

In the given graph $G = \{V, E\}$, each vertex in V is identified with an integer between 1 and $|V|$. We denote the vertex with identification number k as vertex k.

One additional assumption is that in line 5 of DFS and in line 4 of DFS-VISIT, the for loops should iterate over the vertices with their identification numbers in ascending order. For example, in line 5 of DFS, the for loop would iterate with this order: vertex 1, vertex 2, ..., vertex $|V|$.

Input

The first line specifies $|V|$, the number of vertices in G.

The next $|V|$ lines specify the adjacency lists of G. For the k-th line, $1 \leq k \leq |V|$, it starts with an integer specifying the out degree of vertex k, d_k, followed by a sequence of identifcation numbers of the vertices, with consecutive numbers separated by a single space character, $n_1\ n_2\ \ldots\ n_{d_k}$. This specifies that there exist edges $(k, n_1), (k, n_2), \ldots, (k, n_{d_k})$ in G. if the out degree $d_k = 0$, then the line only contains d_k and no other number. You can assume that $n_1\ n_2\ \ldots\ n_{d_k}$ are sorted in ascending order.

Output

The output should have $|V|$ lines. For the k-th line, $1 \leq k \leq |V|$, it starts with the identification number of the vertex, k, a space character, followed by a string of length d_k specifying the classification of the edges coming out from vertex k. That is, if the k-th line of the adjacency list part in the input is $d_k\ n_1\ n_2\ \ldots\ n_{d_k}$, then in the k-th line of the output we have $k\ c_1c_2\ldots c_{d_k}$, with the j-th character c_j of the string representing the classification of the edge (k, n_j), $c_j \in \{T, B, F, C\}$. Here, T represents a tree edge, B represents a back edge, F represents a forward edge, and C represents a cross edge.

Constraint

- $1 \leq |V| \leq 1000$
- There are neither self edges nor multiple edges in G. In other words, you can assume that $0 \leq |E| \leq |V| \times (|V| - 1)$.

Sample Testcases

Sample Input 1

```
5
2 2 4
1 1
3 2 4 5
2 3 5
0
```

Sample Output 1

```
1 TT
2 B
3 CBT
4 TF
5
```

Sample Input 2

```
3
2 2 3
2 1 3
2 1 2
```

Sample Output 2

```
1 TF
2 BT
3 BB
```