

Codeforces 455B – A lot of Games:

Andrew, Fedor and Alex are inventive guys. Now they invent the game with strings for two players.

Given a group of n non-empty strings. During the game two players build the word together, initially the word is empty. The players move in turns. On his step player must add a single letter in the end of the word, the resulting word must be prefix of at least one string from the group. A player loses if he cannot move.

Andrew and Alex decided to play this game k times. The player who is the loser of the i -th game makes the first move in the $(i + 1)$ -th game. Guys decided that the winner of all games is the player who wins the last (k -th) game. Andrew and Alex already started the game. Fedor wants to know who wins the game if both players will play optimally. Help him.

Input

The first line contains two integers, n and k ($1 \leq n \leq 10^5$; $1 \leq k \leq 10^9$).

Each of the next n lines contains a single non-empty string from the given group. The total length of all strings from the group doesn't exceed 10^5 . Each string of the group consists only of lowercase English letters.

Output

If the player who moves first wins, print "First", otherwise print "Second".

Idea:

To solve this problem we need the prefix tree(trie), which will have all the strings from the group. Next we will calculate the two DP:

$\text{win}[u]$ — Can player win if he makes a move now (players have word equal to prefix u in the prefix tree(trie)).

$\text{lose}[u]$ — Can player lose if he makes a move now (players have word equal to prefix u in the prefix tree(trie)).

if u is leaf of trie, then $\text{win}[u] = \text{false}$; $\text{lose}[u] = \text{true}$;

Else $\text{win}[u] = (\text{win}[u] \text{ or } (\text{not } \text{win}[v])); \text{lose}[u] = (\text{lose}[u] \text{ or } (\text{not } \text{lose}[v])),$

such v — children of vertex u .

Let's look at a few cases:

If $\text{win}[u] = \text{false}$, then second player win (first player lose all games).

If $\text{win}[u] = \text{true}$ and $\text{lose}[u] = \text{true}$, then first player win (he can change the state of the game in his favor).

If $\text{win}[u] = \text{true}$ and $\text{lose}[u] = \text{false}$, then if k is odd, then first player win, else second player win.

```
1 // 455B - A Lot Of Games
2 #include<bits/stdc++.h>
3 using namespace std;
4 string s;
5 int cnt,win[100005],lose[100005];
6 struct node{
7     int state,next[26];
8     node(){ state=0; memset(next,-1,sizeof(next)); }
9 }tree[100005];
10
11 void Insert(){
12     int cur = 0;
13     for(int i=0; i<s.size(); i++){
14         int id = s[i]-'a';
15         if(tree[cur].next[id]==-1){
16             tree[cur].next[id] = ++cnt;
17             tree[cnt] = node();
18         }
19         cur = tree[cur].next[id];
20     }
21 }
22
23 void fun(int u){
24     win[u]=0; lose[u]=0; int isleaf = 1;
25     for(int i=0; i<26; i++){
26         int v = tree[u].next[i];
27         if(v!=-1){
28             isleaf = 0;
29             fun(v);
30             win[u] |= !win[v];
31             lose[u] |= !lose[v];
32         }
33     }
34     if(isleaf==1)lose[u]=1;
35 }
36
37 int main(){
38     ios::sync_with_stdio(false); cin.tie(0);
39
40     int n,k; cin>>n>>k;
41     for(int i=1; i<=n; i++){
42         cin>>s;
43         Insert();
44     }
45
46     fun(0);
47
48     if(win[0]==0) printf("Second\n");
49     else if(win[0]==1&&lose[0]==1) printf("First\n");
50     else{
51         if(k&1) printf("First\n");
52         else printf("Second\n");
53     }
54
55     return 0;
56 }
```