```cpp
//
// There is a set A = {a_1, a_2 ...}  consisting of N positive integers.
// Taro and Jiro will play the following game against each other.
// Initially, we have a pile consisting of K stones.
// The two players perform the following operation alternately, starting from Taro:
//    - Choose an element x in A, and remove exactly x stones from the pile.
// A player loses when he becomes unable to play.
//
// Assuming that both players play optimally, determine the winner.
// Time Complexity: O(NK)
//

#include <bits/stdc++.h>
#define ll long long

using namespace std;

inline void umin(int& a, int b) { a = min(a, b); }
inline void umax(int& a, int b) { a = max(a, b); }

int main() {
    int n, k;
    cin >> n >> k;

    vector<int> a(n);
    for (int i = 0; i < n; i++) cin >> a[i];

    vector<vector<int>> dp(k+1, vector<int>({ 1000, -1000 }));
    // player 0 wants to minimize
    // player 1 wants to maximize

    for (int i = 0; i <= k; i++) {
        for (int j = 0; j < n; j++) {
            if (i - a[j] < 0) {
                umin(dp[i][0], 1);
                umax(dp[i][1], 0);
            } else {
                umin(dp[i][0], dp[i-a[j]][1]);
                umax(dp[i][1], dp[i-a[j]][0]);
            }
        }
    }

    cout << (dp[k][0] ? "Second" : "First") << endl;
    return 0;
}
```