```cpp
//
//  There are N children, numbered 1,2,…,N.
//  They have decided to share K candies among themselves.
//  Here, for each i (1≤i≤N), Child i must receive between 0 and a[i] candies (inclusive).
//  Also, no candies should be left over.
//
//  Find the number of ways for them to share candies, modulo 10^9 + 7.
//  Here, two ways are said to be different when there
//  exists a child who receives a different number of candies.
//
//    Time Complexity: O(NK)
//

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

using namespace std;

const int MOD = 1e9 + 7;

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(n, vector<int>(k+1, 0));
    for (int i = k; i >= k - a[0]; i--) dp[0][i] = 1;

    for (int i = 1; i < n; i++) {
        vector<int> pref(k+1);
        pref[k] = dp[i-1][k];
        for (int j = k-1; j >= 0; j--) {
            pref[j] = (pref[j+1] + dp[i-1][j]) % MOD;
        }

        for (int j = 0; j <= k; j++) {
            // dp[i][j] = dp[i-1][j+(0...a[i])]
            dp[i][j] = (pref[j] - (j+a[i]+1 <= k ? pref[j+a[i]+1] : 0) + MOD) % MOD;
        }
    }

    cout << dp[n-1][0] << endl;
    return 0;
}
```