```cpp
//
//  Let N be a positive odd number.
//  There are N coins, numbered 1,2,…,N. For each i (1≤i≤N), when Coin i is tossed,
//  it comes up heads with probability p[i] and tails with probability 1-p[i] tails.
//  Taro has tossed all the N coins. Find the probability of having more heads than tails.
//
//  Time Complexity: O(N^2)
//

#include <bits/stdc++.h>

using namespace std;

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

    // everything in this problem
    // is with 1-based indexing
    // for simplicity
    vector<double> p(n+1);
    for (int i = 1; i <= n; i++) {
        cin >> p[i];
    }

    vector<vector<double>> dp(n+1, vector<double>(n+1, 0));
    dp[1][0] = 1 - p[1];
    dp[1][1] = p[1];

    for (int i = 2; i <= n; i++) {
        for (int j = 0; j <= i; j++) {
            // dp[i][j] - the probability of
            //            getting j heads out of
            //            the coins to the i-th index

            if (j > 0)
                dp[i][j] += dp[i-1][j-1] * p[i];

            dp[i][j] += dp[i-1][j] * (1 - p[i]);
        }
    }

    double ans = 0;
    for (int i = n/2 + 1; i <= n; i++) {
        ans += dp[n][i];
    }

    cout << setprecision(16) << ans << endl;
    return 0;
}
```