```
#include<bits/stdc++.h>
#define ll long long int
using namespace std;

ll getans(string &s, int st, int e, int* visited, vector<int>& dp, int* reach) {
    if(st == e) {
        return 0;
    }
    if(dp[st] != -1) {
        return dp[st];
    }
    if(visited[st] > 2) {
        ll mn = INT_MIN;
        return mn;
    }
    if(s[st] == 'R') {
        ll mn = INT_MAX;
        for(int i = st + 1; (i <= st + reach[st])&&(i <= s.length() + 1); i++) {
            if(visited[i] < 2) {
                visited[i]++;
                ll cost = max((ll)abs(i - st), getans(s, i, e, visited, dp, reach));
                visited[i]--;
                mn = min(cost, mn);
            }
        }
        return dp[st] = mn;
    }
    if(s[st] == 'L' ) {
        ll mn = INT_MAX;
        for(int i = st - 1; (i >= reach[st]) && (i >= 0); i--) {
            if(visited[i] < 2) {
                visited[i]++;
                ll cost = max((ll)abs(st - i), getans(s, i, e, visited, dp, reach));
                visited[i]--;
                mn = min(cost, mn);
            }
        }
        return dp[st] = mn;
    }
}

int main() {
    ll n;
    cin >> n;
    string s;
    cin >> s;
    n = s.length();
    int reach[n+1];
    for(int i = 0; i < s.length(); i++) {
        cin >> reach[i];
    }
    vector<int>dp(n + 3, -1);
    int visited[n+3] = {};
   if(getans(s, 0, n+1 , visited, dp, reach) == INT_MAX) {
       cout << -1 ;
   }
   else {
       cout << getans(s, 0, n+1 , visited, dp, reach);
   }
}
/*

BASIC TC

6
RLRRLR
2 1 2 2 1 1
ANS = 2

7
RRLRRLR
4 100 1 2 0 4 1
ANS = 5

3
LRL
100 2 3
ANS = -1

12
RLLRLRLRLLLL
7 1 1 9 1 8 1 1 5 1 1 1
ANS = 7

16
RLLLLLRLLLRRLRRR
11 5 2 1 2 3 1 7 8 9 1 4 2 0 2 0
ANS = 10


*/
```