

```
1 //
2 // There are N stones, numbered 1,2,...,N. For each i (1≤i≤N), the height of Stone i is h
3 // There is a frog who is initially on Stone 1. He will repeat the following action
4 // some number of times to reach Stone N:
5 // - If the frog is currently on Stone i, jump to one of the following: Stone i+1, i+2 ... i+K.
6 // Here, a cost of | h[i] - h[j] | is incurred, where j is the stone to land on.
7 // Find the minimum possible total cost incurred before the frog reaches Stone N.
8 //
9 // Time Complexity: O(NK)
10 //
11
12 #include <bits/stdc++.h>
13 using namespace std;
14
15 int main() {
16     int n, K;
17     cin >> n >> K;
18
19     vector<int> heights(n);
20     for (int i = 0; i < n; i++) {
21         cin >> heights[i];
22     }
23
24     vector<int> dp(n);
25     dp[0] = 0;
26     for (int i = 1; i < n; i++) {
27         int cur_ans = dp[i-1] + abs(heights[i] - heights[i-1]);
28         for (int k = 2; k ≤ K; k++) {
29             if (i - k ≥ 0) {
30                 cur_ans = min(cur_ans, dp[i-k] + abs(heights[i] - heights[i-k]));
31             }
32         }
33         dp[i] = cur_ans;
34     }
35
36     cout << dp[n-1] << endl;
37     return 0;
38 }
```