// 42-Trapping\_Rain\_Water.cpp //

#include <iostream>

#include <vector>

using namespace std;

class Solution {

public:

// using two pointers

int trap(vector<int>& height)

{

int left = 0, right = height.size() - 1;

int ans = 0;

int left\_max = 0, right\_max = 0;

while (left < right) {

if (height[left] < height[right]) {

printf("height\_left[%d]=%d < height\_right[%d]=%d\n", left, height[left], right, height[right]);

//height[left] >= left\_max ? (left\_max = height[left]) : ans += (left\_max - height[left]);

if (height[left] >= left\_max)

{

left\_max = height[left];

printf("left\_max = %d\n", left\_max);

}

else

{

ans += (left\_max - height[left]);

printf(">>> left\_max=%d, height[left]=%d, inc = %d\n", left\_max, height[left], (left\_max - height[left]));

}

++left;

}

else {

printf("height\_left[%d]=%d >= height\_right[%d]=%d\n", left, height[left], right, height[right]);

//height[right] >= right\_max ? () : ans += (right\_max - height[right]);

if(height[right] >= right\_max)

{

right\_max = height[right];

printf("right\_max = %d\n", right\_max);

}

else

{

ans += (right\_max - height[right]);

printf(">>> right\_max=%d, height[right]=%d, inc = %d\n", right\_max, height[right], (right\_max - height[right]));

}

--right;

}

}

return ans;

}

};

int main()

{

Solution sol;

vector<int> vec = { 0, 1, 0, 2, 1, 0, 1, 3, 2, 1, 2, 1 };//Output : 6

cout << sol.trap(vec) << endl;

}