

CodeCheck Report: trainingFVDSKU-XA8

[Check out Codility training tasks](#)

Test Name:

Summary

Timeline

AI Assistant Transcript

Tasks summary

Task	Time spent	Score
MaxSliceSum	28 min	100%

Total score

100%

Tasks Details

Easy	1. MaxSliceSum	Task Score	Correctness	Performance
	Find a maximum sum of a compact subsequence of array elements.	100%	100%	100%

Task description

A non-empty array A consisting of N integers is given. A pair of integers (P, Q), such that $0 \leq P \leq Q < N$, is called a *slice* of array A. The *sum* of a slice (P, Q) is the total of $A[P] + A[P+1] + \dots + A[Q]$.

Write a function:

```
int solution(vector<int> &A);
```

that, given an array A consisting of N integers, returns the maximum sum of any slice of A.

For example, given array A such that:

```
A[0] = 3  A[1] = 2  A[2] = -6
A[3] = 4  A[4] = 0
```

the function should return 5 because:

- (3, 4) is a slice of A that has sum 4,
- (2, 2) is a slice of A that has sum -6,
- (0, 1) is a slice of A that has sum 5,
- no other slice of A has sum greater than (0, 1).

Write an **efficient** algorithm for the following assumptions:

- N is an integer within the range $[1..1,000,000]$;

Solution

Programming language used:	C++
Total time used:	28 minutes
Effective time used:	28 minutes
Notes:	not defined yet

Task timeline

10:05:2110:33:14

Code: 10:33:13 UTC, cpp, final, score: 100

[show code in pop-up](#)

```
1 // you can use includes, for example:
2 // #include <algorithm>
3 #include <limits>
```

- each element of array A is an integer within the range [-1,000,000..1,000,000];
- the result will be an integer within the range [-2,147,483,648..2,147,483,647].

Copyright 2009–2024 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Test results - Codility

```
4 // you can write to stdout for debugging purposes
5 // cout << "this is a debug message" << endl;
6
7 int solution(vector<int>& A) {
8     int maxslice = INT_MIN;
9     int ending = INT_MIN;
10    bool firstValue = true;
11
12    for (int value : A) {
13        if (firstValue) {
14            ending = value;
15            firstValue = false;
16        } else {
17            ending = std::max(value, ending + value);
18        }
19        maxslice = std::max(ending, maxslice);
20    }
21
22    return maxslice;
23 }
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity: **O(N)**

expand all	Example tests	
▶ example		✓ OK
expand all	Correctness tests	
▶ one_element		✓ OK
▶ two_elements		✓ OK
▶ three_elements		✓ OK
▶ simple		✓ OK
▶ extreme_minimum		✓ OK
▶ fifty_random		✓ OK
▶ neg_const		✓ OK
▶ pos_const		✓ OK
expand all	Performance tests	
▶ high_low_1Kgarbage		✓ OK
▶ 1Kgarbage_high_low		✓ OK
▶ growing_saw		✓ OK
▶ blocks		✓ OK
▶ growing_negative		✓ OK