**7 kyu**

**Maximum Product**

12790% of 294155 of1,240[MrZizoScream](https://www.codewars.com/users/MrZizoScream" \o "This kata's Sensei)

C++

* [TRAIN AGAIN](https://www.codewars.com/kata/maximum-product/train/cpp)
* [NEXT KATA](https://www.codewars.com/trainer/cpp)

Details

[Solutions](https://www.codewars.com/kata/maximum-product/solutions/cpp)

[Discourse (41)](https://www.codewars.com/kata/maximum-product/discuss/cpp)

* Add to Collection
* |
* Share this kata:

Task

***Given*** *an array of integers* , ***Find*** ***the maximum product*** *obtained from multiplying 2 adjacent numbers in the array*.

**Notes**

* ***Array/list*** size is *at least 2* .
* ***Array/list*** numbers could be a *mixture of positives , ngatives also zeros* .

**Input >> Output Examples**

adjacentElementsProduct(new int[] {1, 2, 3}); ==> return 6

***Explanation***:

* ***The maximum product*** *obtained from multiplying* 2 \* 3 = 6, and ***they're adjacent numbers in the array***.

adjacentElementsProduct(new int[] {9, 5, 10, 2, 24, -1, -48}); ==> return 50

***Explanation***:

***Max product*** obtained *from multiplying* 5 \* 10 = 50 .

adjacentElementsProduct(new int[] {-23, 4, -5, 99, -27, 329, -2, 7, -921}) ==> return -14

***Explanation***:

* ***The maximum product*** *obtained from multiplying* -2 \* 7 = -14, and ***they're adjacent numbers in the array***.

[**Playing with Numbers Series**](https://www.codewars.com/collections/playing-with-numbers)

[**Playing With Lists/Arrays Series**](https://www.codewars.com/collections/playing-with-lists-slash-arrays)

[**For More Enjoyable Katas**](http://www.codewars.com/users/MrZizoScream/authored)

ALL translations are welcomed

Enjoy Learning !!

**Zizou**

<https://www.codewars.com/kata/maximum-product/cpp>

1. #include <vector>
2. #include <iostream>
3. #include <stdio.h>
4. #include <limits.h>
6. using namespace std;
8. int adjacentElementsProduct(vector<int> inputArray)
9. {
10. int max = inputArray[0] \* inputArray[1];
11. for(int i=1; i< inputArray.size()-1; i++){
12. int temp = inputArray[i] \* inputArray[i+1];
13. if(temp > max) max = temp;
14. }
15. return max;
16. }
18. int adjacentElementsProduct(vector<int> inputArray)
19. {
20. // Add your code Here .. Enjoy...
21. //return 1 ;
22. int max\_prod = INT\_MIN;
23. for(int i =0; i + 1 < inputArray.size(); i++) {
24. max\_prod = max( max\_prod, inputArray[i] \* inputArray[i+1]);
25. }
27. return max\_prod;
28. }
30. int main() {
32. int arr[] = {9, 5, 10, 2, 24, -1, -48};
33. std::vector<int> v;
34. int size = sizeof(arr)/sizeof(int);
35. for(int i =0; i<size; i++) {
36. v.push\_back(arr[i]);
37. }
39. cout << adjacentElementsProduct(v) <<  endl;

42. return 0;
43. }