**7 kyu**

**Even numbers in an array**

11287% of 18975 of733[Dmitry Kudla](https://www.codewars.com/users/Dmitry%20Kudla)

C++

* [TRAIN AGAIN](https://www.codewars.com/kata/even-numbers-in-an-array/train/cpp)
* [NEXT KATA](https://www.codewars.com/trainer/cpp)

Details

[Solutions](https://www.codewars.com/kata/even-numbers-in-an-array/solutions/cpp)

[Forks (1)](https://www.codewars.com/kata/even-numbers-in-an-array/forks/cpp)

[Discourse (27)](https://www.codewars.com/kata/even-numbers-in-an-array/discuss/cpp)

* Add to Collection
* |
* Share this kata:

Given an array of digitals numbers, return a new array of length number containing the last even numbers from the original array (in the same order). The original array will be not empty and will contain at least "number" even numbers.

For example:

([1, 2, 3, 4, 5, 6, 7, 8, 9], 3) => [4, 6, 8]

([-22, 5, 3, 11, 26, -6, -7, -8, -9, -8, 26], 2) => [-8, 26]

([6, -25, 3, 7, 5, 5, 7, -3, 23], 1) => [6]

<https://www.codewars.com/kata/even-numbers-in-an-array/cpp>

#include <vector>

#include <iostream>

#include <algorithm>

#include <stdio.h>

using namespace std;

std::vector<int> evenNumbers(std::vector<int> arr, int n) {

  //your code here

  std::vector<int> v;

  int cont =0;

  for(int i = arr.size() - 1; i >=0; i-- ) {

        if(cont < n) {

          if(arr[i] % 2 == 0) {

             v.push\_back(arr[i]);

              cont++;

            }

        }else{

            return v;

        }

        //if(cont == n) break;

  }

  std::reverse(v.begin(), v.end());

  return v;

}

int main() {

    std::vector<int> v;

    v.push\_back(8);

    v.push\_back(6);

    v.push\_back(2);

    std::vector<int> res = evenNumbers(v, 0);

    for(int i =0; i<res.size(); i++) {

        cout << res[i] << endl;

    }

    return 0;

}