Largest Element

Implement a function that takes array of integers as input and returns the largest element.

Sample Input

-3 4 1 2 3

-1 -2 -3 -3 8

Sample Output

4

8

Solution: largestElement.cpp

Maximum Subarray Sum

Implement a function that takes an array as input and returns the maximum subarray sum.

Sample Input

```
1 -2 3 4 4 -25 0 -1 0 1 2 -1
```

Sample Output

```
117
```

Solution: maxSubArraySum.cpp

Lower Bound

Given an array of integers A (sorted) and a integer Val.

Implement a function that takes A and Val as input parameters and returns the lower bound of Val.

Note : If Val is not present in array then Lower bound of a given integer means integer which is just smaller than given integer.

Otherwise Val itself is the answer.

If Val is less than smallest element of array A then return '-1' in that case.

Example 1 :

A = [-1, -1, 2, 3, 5]

Val = 4

Answer : 3

Since 3 is just smaller than 4 in the array.

Example 2 :

A = [1, 2, 3, 4, 6]

Val = 4

Answer : 4
Since 4 is equal to 4

Solution: lowerBound.cpp

Sorted Pair Sum

Given a sorted array and a number x, find a pair in array whose sum is closest to x.

Input Format

In the function an integer vector and number x is passed.

Output Format

Return a pair of integers.

Sample Input

```
{10, 22, 28, 29, 30, 40}, x = 54
```

Sample Output

```
22 and 30
```

Solution: sortedPairSum.cpp

K-Rotate

Given an integer vector and a value k, your task is to rotate the array k times clockwise.

Input Format

In the function an integer vector and number k is passed.

Output Format

Return an integer vector.

Sample Input

```
{1, 3, 5, 7, 9}, x = 2
```

Sample Output

```
{7, 9, 1, 3, 5}
```

Solution: kRotate.cpp

Minimum Difference

Implement a function that takes in two non-empty arrays of integers, finds the pair of numbers (one from each array) who absolute difference is closest to zero, and returns a pair containing these two numbers, with the first number from array. Only one such pair will exist for the test.

Input

```
Array1 = [23, 5, 10, 17, 30]Array2 = [26, 134, 135, 14, 19]
```

Output

```
17,19
```

Solution: minDiff.cpp

Array Products

Implement a function that takes in a vector of integers, and returns a vector of the same length, where each element in the output array is equal to the product of every other number in the input array. Solve this problem **without using division.**

In other words, the value at output[i] is equal to the product of every number in the input array other than input[i]. You can assume that answer can be stored inside int datatype and no-overflow will occur due to products.

Sample Input

Both inputs and outputs are vectors.

```
{1,2,3,4,5}
```

Sample Output

```
{120, 60, 40, 30, 24}
```

Solution: arrayProducts.cpp

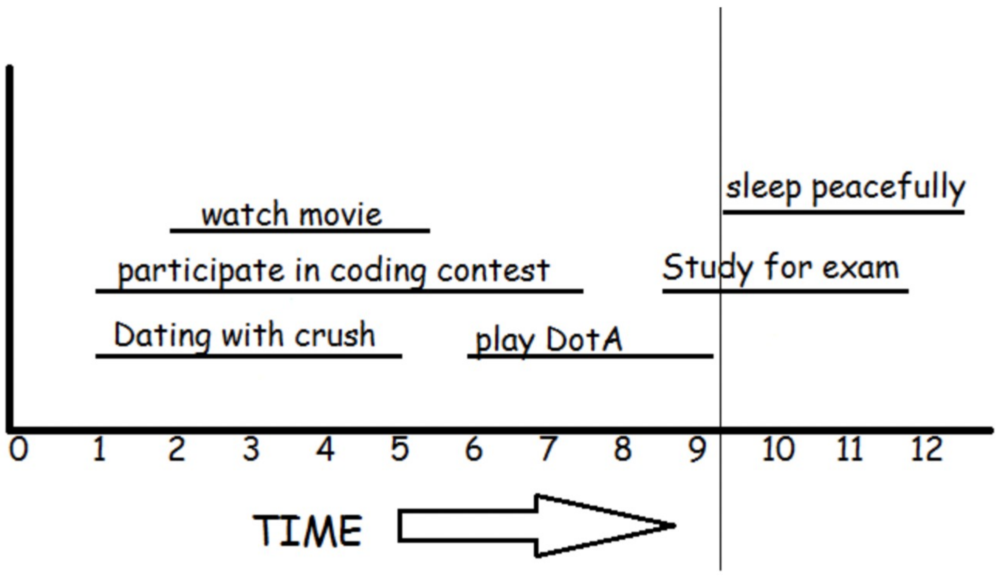
Busy Life

You are actually very busy person. You have a long list of activities. Your aim is to finish much as activities as possible.

In the given figure, if you go to date with crush, you cannot participate in the coding contest and you can't watch the movie. Also if you play DotA, you can't study for the exam. If you study for the exam you can't sleep peacefully. The maximum number of activities that you can do for this schedule is 3.

Either you can

- watch movie, play DotA and sleep peacefully (or)
- date with crush, play DotA and sleep peacefully



Given the start and finish times of activities, print the maximum number of activities. Input is already store in a vector of pairs. Each pair contains the start and end of the activity.

Sample Input

(7,9) (0,10) (4,5) (8,9) (4,10) (5,7)

Output
3

Expected Complexity

O(NLogN)

Solution: busyLife.cpp