

# DevsuCodeJam 2011

## Questions

### Piece of Cake - 12 questions - 1 point per question

#### 1. Let's start ...

Look at this series: 7, 6, 8, 4, 9, 2, 10, 0, 11, -2, ...

Create a function that receives two integers: x and y. If any of them are 0 or negative, or if they are greater than 255, the function should return -1

Otherwise, the function should return the sum of the x and y elements of the series.

For example: If the function receives x=1, y=3, it should return: 15. (Because the sum of the first plus the third argument is 7+8=15). If the function receives x=8, y=9, it should return 11. (Because the sum of the 8th plus the 9th element is 0+11=11).

The function will receive 2 integers, return an integer.

#### 2. Wow, this will be easy....

Look at this series: 2, 2, 4, 12, 48, ... the seed for this series was the number 2.

Look at this series: 3, 3, 6, 18, 72, ... the seed for this series was the number 3.

Create a function that receives two integers: x, and y. If any of them are 0 or negative, or if they are greater than 255, the function should return -1

The function should return the y element of the series generated by x.

For example, if the series receives x = 3, y = 4, it should return 72, because 72 is the 4th element of the series generated when x = 3.

The function will receive 2 integers, and return an integer.

### 3. That was pretty easy, give me another...

Look at this series: 60, 30, 20, 15, 12... the seed for this series was the number 60.

Create a function that receives two integers: x, and y. If any of them are 0 or negative, or if they are greater than 255, the function should return -1.

The function should return the y element of the series generated by x.

For example: If the function receives x=60, y=3, it will return 20, because 20 is the 3th element in the series generated when x = 60.

The function will receive 2 integers, return a floating point value.

### 4. Remove strings? Piece of cake

Given two strings S1 and S2. Delete from S1 all those characters which occur in S2. Return a clean S1 with the relevant characters deleted. Any character deletes both uppercase and lowercase.

For example, given:

S1 = "DevsuCodeJam is just great!"

S2 = "I am here! :)"

The function should return: "DvsuCodJsjustgt".

The function will receive 2 strings and return a string

### 5. Just One Line

Write a function to remove duplicates from a sorted array of integers. Pretty easy, right? What about making it in one line of code? (You can use as many statements as needed, but the code should be written in one line).

Example:

If the function receives this array:

A = [-3, -2, 0, 0, 5, 7, 9, 11, 11, 25]

The function should return:

A = [-3, -2, 0, 5, 7, 9, 11, 25]

The function will receive an array of integers, and return an array of integers.

### 6. We want more!

Given a string, which contains words and spaces (no special characters), create a function that return a string with the words in a reverse order.

Example:

If the function receives: "this is a test", it should return: "test a is this".

If an empty string is received, an empty string should be returned. If only one word received, the same word should be returned.

The function will receive a string and return a string.

## **7. Lowercase**

Given a string which contain letters (uppercase and lowercase), numbers, and special characters, return the same string in all lowercase.

For example, if the function receives:

"Ñañito, QUÉ bien! THIS is a sample text, Lorem Ipsum, 2 Be Converted."

The function should return:

"ñañito, qué bien! this is a sample text, lorem ipsum, 2 be converted."

The function should consider converting: All characters from A-Z, Á,É,Í,Ó,Ú and Ñ. Other characters will remain the same.

### **Limitation:**

The conversion should be done considering the ASCII values. Obviously you CAN'T use the functions provided by the language (toLowerCase(), lowercase(), etc.). You CAN'T have a huge switch statement with cases for each letter, or lots of if/else statements.

This function will receive an string and return an string

## 8. A-Words

Given a string, find the number of words that has at least one "a" character (uppercase or lowercase). Do not take into account character variations like á, à, etc... only the simple "a" and "A" counts.

The words are always separated by a space, a comma, a semicolon or a dot.

For example:

If the function receives: "this is a sample text, it has a lot of analysis." The function should return 5, since five words has "a" characters. (a, sample, has, a, analysis).

The function will receive a string and return an integer.

**Limitations:** Do not use the `split()` function, or similar.

## 9. The power of two

Given a positive integer number determine if it's the power of two of another integer.

Don't start coding, read the limitations.

For example:

If the function receives 25, it should return TRUE, because  $5^2 = 25$

If the function receives 1, it should return TRUE, because  $1^2 = 1$

If the function receives 16, it should return TRUE, because  $4^2 = 16$

If the function receives 14, it should return FALSE.

**Limitation:** You CAN'T use functions of square roots (`sqrt()` or similar), potentiation (`pow()` or similar). ONLY the basic arithmetic operations (sum, subtraction, multiplication, division), and any logic operations are allowed.

The function receives a positive integer greater than 0, and should return a boolean value.

## 10. Perfect numbers

A perfect number is a positive integer that is equal to the sum of its proper divisors. For example, 6 is a perfect number because  $6=1+2+3$ .

Create a function that receives two values X and Y and return the smaller perfect number found, which is greater or equal than X and lower or equal than Y. If no perfect number found, it should return -1.

For example, if the function receives  $X=5$ ,  $Y=7$ , it should return 6, because 6 is the smaller perfect number between 5 and 7.

The function will receive two integers and return one integer.

## 11. Counter

Given an array of integers, find which is repeated more times. Return the number that has more repetitions. If two numbers has the same amount of repetitions, return the lower number.

For example, given this array:

$A = [1, 5, 3, -2, 4, 2, 4, -2, 5, 5, 2, 1, 3]$

1 is repeated 2 times

5 is repeated 3 times

3 is repeated 2 times

-2 is once

4 is repeated 2 times

2 is repeated 2 times

The most repeated number is 5. The function should return: 5. (Because 5 is repeated 3 times in the array).

The function will receive an array of integers and return an integer.

## 12. The last piece of cake

Equilibrium index of a sequence is an index such that the sum of elements at lower indexes is equal to the sum of elements at higher indexes.

Create a function that receives an array of integers and returns the first equilibrium index found. If no equilibrium index found, the function should return -1

For example, if the array received is:

$A = [-7, 1, 5, 2, -4, 3, 0]$

3 is an equilibrium index, because:

$$A[0] + A[1] + A[2] = A[4] + A[5] + A[6]$$

In other words, you should find the index of the array where the sum of the left elements is equal to the sum of the right elements.

In the example, the function will return 3, because it's the first equilibrium index found in the array.

The function receives an array of integers and return an integer.

## Easy - 6 questions - 2 points per question

### 13. Something easy...

You are given an array with positive and negative integers. Write a function to change the elements order in the array such that negative integers are at the beginning, the positive integers are at the end. Zero (0) and integers that have the same sign don't change order.

For example, if the function receives:

```
a[0] = 4;  
a[1] = -3;  
a[2] = -100;  
a[3] = 7;  
a[4] = 0;  
a[5] = 1;  
a[6] = -6;
```

the function should return:

```
a[0] = -3;  
a[1] = -100;  
a[2] = -6;  
a[3] = 4;  
a[4] = 7;  
a[5] = 0;  
a[6] = 1;
```

The function receives an array of integers and return an array of integers.

#### Limitations:

You CAN'T use sorting methods provided by the language. (eg. `Array.sort()`, `sort()`, etc...). If you need to, you should create your own implementation of the sorting function.

#### **14. Okay, okay okay, ... is this a programming contest or a “series” test?**

Look at this series: 53, 35, 64, 46, 75, 57, 86, 68, 97, 79, 108, 810, 119, 911, 1210, 1012, ... the seeds for this series were the numbers 5 and 3.

Look at this series: 103, 310, 114, 411, 125, 512, 136, 613, 147, 714, 158, 815, 169, 916, 1710, 1017, ... the seeds for this series were the numbers 10 and 3.

Look at this series: 1012, 1210, 1113, 1311, 1214, 1412, 1315, 1513, 1416, 1614, 1517, 1715, 1618, 1816, 1719, 1917, ... the seeds for this series were the numbers 10 and 12.

Create a function that receives three integers: x, y and z. If any of them are 0 or negative, or if they are greater than 255, the function should return -1

The function should return the z element of the series generated by x and y.

For example: If the function receives x=5, y=3, z=3, it will return 64, because 64 is the 3th element in the generated series when x=5 and y=3.

A (big) clue: The second number is just the first number, with it's parts inverted; the fourth number is just the third number, with it's parts inverted, and so forth...

The function will receive 3 integers, return an integer.



## 15. Contiguous subarrays

Given an array of integers (positive and negative), find the contiguous subarray with the largest sum. Return the sum.

The subarray can be of any length, even it could be the whole array. It could be also a single element.

For example:

If the function receives the following array:

$A=[4, -3, 7, 2, 4, -5, 1, 2]$

Some contiguous subarrays are:

Subarray from index 0 to 1:  $[4, -3]$  sums 1

Subarray from index 2 to 4:  $[7, 2, 4]$  sums 13

Subarray from index 0 to 7:  $[4, -3, 7, 2, 4, -5, 1, 2]$  sums 12

Subarray from index 0 to 0:  $[4]$  sums 4

....

$[4, 7, 2, 4]$  is NOT a contiguous subarray.

The function will find that the contiguous subarray with the largest sum is  $[4, -3, 7, 2, 4]$ , which sums 14. The function should return 14.

## 16. Series again

Look at this series: 3, 6, 24, 144, ... the seed for this series was the number 3.

Look at this series: 8, 16, 64, 384, ... the seed for this series was the number 8.

Create a function that receives three integers: x, y and z. If any of them are 0 or negative, or if they are greater than 255, the function should return -1

This function should return a value calculated based on y and z from the series generated by x. (x is the seed for the series).

For example, if the function receives x = 3, y = 1, z = 3, the function will find (based on x), that the series is 3, 6, 24, 144, ..., based on that series, and based on y=1 and z=3, the function should return 33.

If the function receives x = 8, y = 2, z = 4, the function should return 464. If the function receives x = 5, y = 2, z = 2, the function should return 10. For any case where y > z, the function will return 0.

To find the relationship between y, z and the result is part of the problem to be resolved.

The function will receive 3 integers, and return an integer.

## 17. Words to numbers

Create a function that transform a string into a number. The number can be between 0 and 255.

For example:

If the function receives "zero", it should return 0.

If the function receives "One", it should return 1.

If the function receives "eleven", it should return 11.

If the function receives "thirteen", it should return 13.

If the function receives "fifty five", it should return 55.

If the function receives "ONE HUNDRED", it should return 100.

If the function receives "Two hundred thirty one", it should return 231.

If the function doesn't understand the string, it should return -1

The function will receive a string and return an integer.

## 18. Sorting, the geeks way,...

Create a function that sorts an array of words in alphabetical order. The text will always be lowercase, and won't contain any special character or number. Do not use sorting functions provided by the language (read the limitations).

For example:

If the function receives:

```
A = ['test', 'contest', 'programming', 'more'];
```

The function should return:

```
['contest', 'more', 'programming', 'test']
```

**Limitations:** We want to make it interesting. Do not use any sorting function (sort, Array.sort(), etc.) provided by the language. You can create your own sorting function.

The function will receive an array of strings and return an array of strings.

## Medium - 4 questions - 3 points per question

### 19. Ok, let's the party begin...

Create a function that receives a 32 bit integer and return the number of ones in the binary of that number. (caution: looping through testing each bit is not a solution).

The function can receive any positive 32 bit integer.

For example:

If the function receives 25, it should return 3. Why? Because 25 in binary is 11001, which has 3 ones.

The function will receive an integer and return an integer.

### 20. Come on, give me something challenging man!

From Wikipedia:

In number theory, a **partition** of a positive integer  $n$ , also called an **integer partition**, is a way of writing  $n$  as a sum of positive integers. Two sums that differ only in the order of their summands are considered to be the same partition.

The partitions of 4 are listed below:

1. 4
2. 3 + 1
3. 2 + 2
4. 2 + 1 + 1
5. 1 + 1 + 1 + 1

The partitions of 8 are listed below:

1. 8
2. 7 + 1
3. 6 + 2
4. 6 + 1 + 1
5. 5 + 3
6. 5 + 2 + 1
7. 5 + 1 + 1 + 1
8. 4 + 4
9. 4 + 3 + 1
10. 4 + 2 + 2
11. 4 + 2 + 1 + 1
12. 4 + 1 + 1 + 1 + 1
13. 3 + 3 + 2

- 14.  $3 + 3 + 1 + 1$
- 15.  $3 + 2 + 2 + 1$
- 16.  $3 + 2 + 1 + 1 + 1$
- 17.  $3 + 1 + 1 + 1 + 1 + 1$
- 18.  $2 + 2 + 2 + 2$
- 19.  $2 + 2 + 2 + 1 + 1$
- 20.  $2 + 2 + 1 + 1 + 1 + 1$
- 21.  $2 + 1 + 1 + 1 + 1 + 1 + 1$
- 22.  $1 + 1 + 1 + 1 + 1 + 1 + 1 + 1$

Create a function that receives one integer:  $x$ . If the value received is 0 or negative, or if it is greater than 255, the function should return -1

The function should return the number of partitions  $x$  has.

For example: If the function receives 4, it should return 5. If it receives 8, it will return 22, etc.

The function will receive one integer, return another integer.

## 21. Spirals!

Given a 2-D matrix of characters, write a function that returns a string with the characters from the matrix in spiral order. (Starting from  $[0][0]$  element, next element:  $[0][1]$ , etc... ).

The matrix can have any length, but it will always be a square.

For example: Given this matrix:

```
a b c
d e f
g h i
```

Return a string with the value: "abcfihgde".

## 22. Spirals again :)

Given a 2-D matrix of integers, write a function that, in spiral order, sums and then subtracts and then multiply the values of the elements. (Starting from  $[0][0]$  element, next element:  $[0][1]$ , etc... )

The matrix can have any length, it will always be a rectangle, but not necessarily a square.

For example: Given this matrix:

```
5 4 7
1 2 3
3 2 1
```

We start with the first element: 5

Then we sum the next element:  $5+4 = 9$

Then we subtract the next element:  $9 - 7 = 2$

Then we multiply the next element:  $2 * 3 = 6$

Then we sum the next element:  $6 + 1 = 7$

Then we subtract the next element:  $7 - 2 = 5$

Then we multiply the next element:  $5 * 3 = 15$

Then we sum the next element:  $15 + 1 = 16$

Then we subtract the next element:  $16 - 2 = 14$

The function should return 14

## Difficult - 2 questions - 6 points per question

### 23. Postfix (our flavor)

Postfix expressions are arithmetical expressions where the operators come after anything they operate on. Postfix is important because it maintains precedence without the use of (), and they're "easy" to evaluate.

Create a function that receives a string with a postfix expression and return the computed value.

The function will receive a correct postfix expression. There will be one space between each number/operator. The operators are + (addition), - (subtraction), \* (multiplication), power (\*\*). The negative numbers are preceded by a minus sign (-). Do not consider the case of a number raised to a negative power (exponent will always be positive).

Examples of expected results:

Input	Output
5 3 *	15
6 4 * 2 +	26
10 25 + 3 * 100 50 * - 85 + 10 13 - 4 * 36 * +	-5242
10 -2 + 5 *	40
10 2 **	100

## 24. Labyrinth

Given a 2-D matrix of boolean values, find the number of steps required to exit from the top left corner to the bottom right corner of the matrix. The FALSE values will be considered as walls, and the TRUE values as empty spaces where we can walk.

The top left corner is the [0][0] value of the matrix. The bottom right corner is the [n][m] value of the matrix. A matrix will always be rectangular, but not always a square.

You are allowed to move in any direction (up, down, left, right, diagonal).

If there's no a path, the function should return -1.

For example:

Given the matrix: (T is true, F is false)

```
T F F F T
F T T T F
F F T F T
T F T T T
```

The shortest path will take 4 steps:

1. From [0][0] to [1][1]
2. From [1][1] to [2][2]
3. From [2][2] to [3][3]
4. From [3][3] to [3][4]

The function should return number 4.

Given the matrix:

```
T T T T T F
F F F T T F
F T T F F T
T F F F F F
T T T T T T
```

The shortest path will take 11 steps. The function should return the number 11.