Answers by : benson Bundi k.

[beunra@yahoo.com](mailto:beunra@yahoo.com)

1. in python

import numbers

def checkNumbers(num1, num2):

#only evaluate number types

if not isinstance(num1, numbers.Number) or not isinstance(num2, numbers.Number):

return "false";

highest = max(num1, num2)

if num1==num2:

return "-1";

elif highest == num2:

return "true";

else:

return "false";

numOne = input("Enter first number: ")

print "You entered: {}".format(numOne)

numTwo = input("Enter second number: ")

print "You entered: {}".format(numTwo)

print "Is second number larger: {} ".format(checkNumbers(numOne,numTwo))

2.

process.stdin.resume();

process.stdin.setEncoding('ascii');

var input\_stdin = "";

var input\_stdin\_array = "";

var input\_currentline = 0;

process.stdin.on('data', function (data) {

input\_stdin += data;

});

process.stdin.on('end', function () {

input\_stdin\_array = input\_stdin.split("\n");

main();

});

function readLine() {

return input\_stdin\_array[input\_currentline++];

}

/////////////// ignore above this line ////////////////////

function main() {

var n = parseInt(readLine());

console.log(binarySearch(n));

}

function isFailedTest(version){

var status =false;

if(version > 8 )status = true;

console.log(`Testing version #${version} isFailedTest: ${status} `);

return status;

}

//returns the lowest failed version or null if no version has failed

//to minimise api calls, binary search used

function binarySearch(value) {

var lo = 1, hi = value, mid,lowest=null;

while (lo <= hi) {

mid = Math.floor((lo+hi)/2);

if (isFailedTest(mid)){

hi = mid - 1;

lowest = mid;

} else{

lo = mid + 1;

}

console.log(`lo: ${lo} mid: ${mid} hi: ${hi} lowest Failed : ${lowest}`);

}

return lowest;

}

3.

//http://mowson.org/karl/2016/2016-05-20\_alpinelinux\_vm\_under\_virtualbox/

process.stdin.resume();

process.stdin.setEncoding('ascii');

var input\_stdin = "";

var input\_stdin\_array = "";

var input\_currentline = 0;

process.stdin.on('data', function (data) {

input\_stdin += data;

});

process.stdin.on('end', function () {

input\_stdin\_array = input\_stdin.split("\n");

main();

});

function readLine() {

return input\_stdin\_array[input\_currentline++];

}

/////////////// ignore above this line ////////////////////

function main() {

//sort in ascending order

var greedyChild = readLine().split(',').map(Number).sort((a,b)=>a-b);

var sweets = readLine().split(',').map(Number).sort((a,b)=>a-b);

console.log(findMatches(greedyChild,sweets));

}

function findMatches(greedyChild,sweets){

var matches = 0;

for (var x = 0; x < greedyChild.length ; x++){

for (var y = 0; y < sweets.length ; y++){

if(greedyChild[x] <= sweets[y]){

console.log(`found at greedyChild index ${x} sweets index ${y} match :${greedyChild[x]} <= ${sweets[y]} `);

matches++;

sweets = sweets.slice(y+1);

break;

}

}

}

return matches;

}

4.

http://mowson.org/karl/2016/2016-05-20\_alpinelinux\_vm\_under\_virtualbox/

process.stdin.resume();

process.stdin.setEncoding('ascii');

var input\_stdin = "";

var input\_stdin\_array = "";

var input\_currentline = 0;

process.stdin.on('data', function (data) {

input\_stdin += data;

});

process.stdin.on('end', function () {

input\_stdin\_array = input\_stdin.split("\n");

main();

});

function readLine() {

return input\_stdin\_array[input\_currentline++];

}

/////////////// ignore above this line ////////////////////

function main() {

//sort in ascending order

var prices = readLine().split(',').map(Number);

console.log(findMax(prices));

}

function findMax(prices){

console.log(`input: ${prices}`);

var profit = 0;

for(var i=1; i<prices.length; i++){

var diff = prices[i]-prices[i-1];

if(diff > 0){

profit += diff;

console.log(`add profit (${prices[i]} - ${prices[i-1]}) ${diff}`);

}

}

return profit;

}

Instructions

* Attempt all questions
* Marks are awarded based on the following:
  + Simplicity
  + Readability
  + Maintainability
  + Resource optimization
  + Functionality
  + Security
  + Regulatory requirements

**Question 1**

Function checkNumbers(number1, number2) takes in 2 numbers and returns a string **true** if number2 is greater than number1, otherwise it returns String **false.** If the parameters are equal to each other it returns the string -1.

**Using Python, write the function checkNumbers(num1, num2)**

Sample Test Cases

**Input:50 & num2 = 100**

**Output:"true"**

**Input:50 & num2 = 50**

**Output:"-1"**

Question 2:

As Senior Software Engineer, you lead a team of Software Engineers to develop a new software and you use Semantic versioning to version your code. Each of the version is based on a previous version such that all versions after the bad version are also bad.

Supposing that you have n versions [1,2,3,4,5,…..,n] you want to write a software that finds the first bad version, which makes all the versions that follow bad.

In order to determine the bad version, we have an API bool isFailedTest(version) which tells you whether the version is bad or not (if it fails or passes the test respectively).

**Write a function that checks for the first bad version**. **You should minimize the number of calls to the API**

**Question 3**

As a [good] parent ☺, you want to give your children some sweets and you can only give each child at most one sweet. Each sweet ***j*** has a size ***sj***. Each child ***i*** has a greed factor ***gi*** which is the minimum size of a sweet that the child will be contented with. If ***sj>=gi****,* we can assign the sweet **j** to the child and the child will be content. **Write a function that will maximize the number of your children that will be content and output the maximum number that will be content.**

* Assume that the greed factor is always a positive value.
* You cannot assign more than one cookie to one child.

**Example**

You have 3 children: x, y and z with greed factor [1,2,3], and you have 2 cookies with sizes [1,1].

**Input:[1,2,3], [1,1]**

**Output:1**

Explanation: Though there are 2 cookies their sizes are the same (1) and therefore only one child (whose greed factor is 1) will get contented. You need to output 1.

**Question 4:**

Say you have an array for which the ith element is the price of a given stock on day i.

Design an algorithm to find the maximum profit. You may complete as many transactions as you like (ie, buy one and sell one share of the stock multiple times). However, you may not engage in multiple transactions at the same time (ie, you must sell the stock before you buy again).

Question 5

You are asked to cut off trees in a forest for a golf event. The forest is represented as a non-negative 2D map, in this map:

* 0 represents the obstacle can't be reached.
* 1 represents the ground can be walked through.

The place with number bigger than 1 represents a tree can be walked through, and this positive number represents the tree's height.

You are asked to cut off all the trees in this forest in the order of tree's height - always cut off the tree with lowest height first. And after cutting, the original place has the tree will become a grass (value 1).

You will start from the point (0, 0) and you should output the minimum steps you need to walk to cut off all the trees. If you can't cut off all the trees, output -1 in that situation.

You are guaranteed that no two trees have the same height and there is at least one tree needs to be cut off.

**Example 1:**

Input:

[

[1,2,3],

[0,0,4],

[7,6,5]

]

Output: 6

**Example 2:**

Input:

[

[1,2,3],

[0,0,0],

[7,6,5]

]

Output: -1

Example 3:

Input:

[

[2,3,4],

[0,0,5],

[8,7,6]

]

Output: 6

Explanation: You started from the point (0,0) and you can cut off the tree in (0,0) directly without walking.

Hint: size of the given matrix will not exceed 50x50.