JAVA EVAL()

“ 5+5” 10

“alert(“welcome”)”

Class k1

{

Class k2

{

}

}

Class

Class car

{

Carno;

Car model

}

Class kiya extends car

{

Car series

}

Kiya k=new kiya();

Car c=new car();

Kiya k1=new car();

Class car

{

Cno

Cname

Static Class bike

{

Bnu;

Bname;

}

}

Car.bike bc=new Car.bike();

JavaScript strict mode

The strict mode in JavaScript does **not** allow following things:

1. Use of undefined variables
2. Use of reserved keywords as variable or function name
3. Duplicate properties of an object
4. Duplicate parameters of function
5. Assign values to read-only properties
6. Modifying arguments object
7. Octal numeric literals
8. with statement
9. eval function to create a variable

promises:

**Promises** are used to handle asynchronous operations in JavaScript. They are easy to manage when dealing with multiple asynchronous operations where callbacks can create callback hell leading to unmanageable code.

* **A Promise has four states:**
  1. **fulfilled**: Action related to the promise succeeded
  2. **rejected**: Action related to the promise failed
  3. **pending**: Promise is still pending i.e. not fulfilled or rejected yet
  4. **settled**: Promise has fulfilled or rejected

<!--🡪

//

//\* \*/

CODING QUESTIONS :

**Problem Statement-:**Lets define Ternary numbers to be the positive integer numbers that consist of digits 0,1,2 only. Let’s also define the operation TOR(x,y) to be the following operation on two Trenary numbers x and y (its output is z = TOR(x,y)).

1. Go through the digits one by one .
2. For each digit i, the digit i in the output will be z[i] = (x[i] + y[i]) % 3

You are given a positive integer n, and a long positive integer c, your job is to find two Ternary   numbers a,b (of length n) where c = TOR(a,b) and max(a,b) is as minimum as possible. After finding that pair output max(a,b). If there’s no such pair, output -1.

* Note 1: the given numbers won’t have any leading zeros.
* Note 2: a,b should be of length n without leading zeros.
* Note 3: since the answer can be very large, output it modulo 1000000007 (10^9 +7).

**Function Description:**

Complete the TOR function in the editor below. It has the following parameters(s):

Parameters:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| n | integer | The length of a and b |
| c | string | The number which should satisfy the equation c=TOR(a,b). |

The function must return an INTEGER denoting the max(a,b) when it’s minimized.

**Problem Statement-:**Lets define Ternary numbers to be the positive integer numbers that consist of digits 0,1,2 only. Let’s also define the operation TOR(x,y) to be the following operation on two Trenary numbers x and y (its output is z = TOR(x,y)).

1. Go through the digits one by one .
2. For each digit i, the digit i in the output will be z[i] = (x[i] + y[i]) % 3

You are given a positive integer n, and a long positive integer c, your job is to find two Ternary   numbers a,b (of length n) where c = TOR(a,b) and max(a,b) is as minimum as possible. After finding that pair output max(a,b). If there’s no such pair, output -1.

* Note 1: the given numbers won’t have any leading zeros.
* Note 2: a,b should be of length n without leading zeros.
* Note 3: since the answer can be very large, output it modulo 1000000007 (10^9 +7).

**Function Description:**

Complete the TOR function in the editor below. It has the following parameters(s):

Parameters:

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| n | integer | The length of a and b |
| c | string | The number which should satisfy the equation c=TOR(a,b). |

The function must return an INTEGER denoting the max(a,b) when it’s minimized

**Input Format**

* The first line contains an integer, n, denoting the length of a and b .
* The next line contains a string, c, denoting the number which should satisfy the equation c=TOR(a,b).

**Sample Cases:**

* **Sample input 1**   
  2                                              
  22
* **Sample output**11

### **Question 2 : Special LIS**

**Problem statement-:**Given an array of N integers. Find the length of the longest increasing subsequence such that the difference between adjacent elements of the longest increasing subsequence is also increasing.

Note: It’s not required that the sequence should have strictly increasing numbers and the differences are also not required to be strictly increasing.

Ex: [1,2,3,4,5]. The longest increasing subsequence will be [1,2,4,5] but the difference array in this case is [1,2,1] which is not increasing. If we consider [1,2,4] or [1,2,5] then the numbers as well as the differences are in increasing order.

**Function Description:**

Complete the specialLIS function in the editor below. It has the following parameter(s):

Parameters:

| **Name** | **Type** | **Type** |
| --- | --- | --- |
| size | Integer | size of array |
| arr | Integer array | array |

**Return**:

The function must return an INTEGER denoting the length of the longest increasing sub sequence with increasing differences.

**Constraints:**

* 1 <= size <= 500
* 1 <= arr[i] <= 10^5

**Input Format:**

* The first line contains integer, size, denoting the size of the array.
* Each line i of the size subsequent lines (where 0 <= i < size) contains an integer describing the ith elements of the array.

**Sample Cases**

* **Sample Input 1**5  
  1  
  2  
  3  
  4  
  5
* **Sample Output 1**5