https://www.hackerrank.com/challenges/utopian-tree/problem?isFullScreen=true

HackerRank Prepare > Algorithms > Implementation > Utopian Tree

Exit Full Screen View

Problem | Submissions | Leaderboard | Discussions | Editorial

The Utopian Tree goes through 2 cycles of growth every year. Each spring, it doubles in height. Each summer, its height increases by 1 meter.

A Utopian Tree sapling with a height of 1 meter is planted at the onset of spring. How tall will the tree be after n growth cycles?

For example, if the number of growth cycles is $n = 5$, the calculations are as follows:

Period	Height
0	1
1	2
2	3
3	6
4	7
5	14

Function Description

Complete the utopianTree function in the editor below.

utopianTree has the following parameter(s):

- int n: the number of growth cycles to simulate

Returns

- int: the height of the tree after the given number of cycles

Input Format

The first line contains an integer, t, the number of test cases.

t subsequent lines each contain an integer, n, the number of cycles for that test case.

Constraints

$1 \leq t \leq 10$

$0 \leq n \leq 60$

Sample Input

```
3
0
1
4
```

Change Theme | Language: Java 7

```
1  import java.util.Scanner;
2
3  public class Solution {
4      public static void main(String[] args) {
5          final int
6              MONSOON = 100,
7              SUMMER = 200;
8          Scanner input = new Scanner(System.in);
9          int caseCount = Integer.parseInt(input.nextLine());
10         for (int i = 0; i < caseCount; i++) {
11             int treeHeight = 1;
12             int cycleType = MONS
13             int cycleCount = Int
14             for (int j = 0; j < cycleCount; j++) {
15                 switch (cycleType) {
16                 case MONSOON:
17                     treeHeight = treeHeight * 2;
18                     cycleType = SUMMER;
19                     break;
20                 case SUMMER:
21                     treeHeight += 1;
22                     cycleType = MONSOON;
23                     break;
24                 }
25             }
26             System.out.println(treeHeight);
27         }
28     }
29 }
30
```

int cycleCount - Solution.main(String[])

Line: 30 Col: 1

Upload Code as File | Test against custom input | Run Code | Submit Code

30°C Haze | Search | ENG IN | 8:41 AM 7/12/2023

https://www.hackerrank.com/challenges/triple-sum/problem?h_r=internal-search

Change Theme Language Python 3

```
    bi = 0
    ci = 0

    ans = 0

    while bi < len(b):
        while ai < len(a) and a[ai] <= b[bi]:
            ai += 1

        while ci < len(c) and c[ci] <= b[bi]:
            ci += 1

        ans += ai * ci
        bi += 1

    return ans

if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')

    lenaLenbLenc = input().split()

    lena = int(lenaLenbLenc[0])

    lenb = int(lenaLenbLenc[1])

    lenc = int(lenaLenbLenc[2])

    arra = list(map(int, input().rstrip().split()))

    arrb = list(map(int, input().rstrip().split()))

    arrc = list(map(int, input().rstrip().split()))

    ans = triplets(arra, arrb, arrc)

    fptr.write(str(ans) + '\n')

    fptr.close()
```

Line: 54 Col: 17

Upload Code as File Test against custom input Run Code Submit Code

30°C Haze Search ENG IN 8:49 AM 7/12/2023

https://www.hackerrank.com/challenges/separate-the-numbers/problem?isFullScreen=true

HackerRank Prepare > Algorithms > Strings > Separate the Numbers

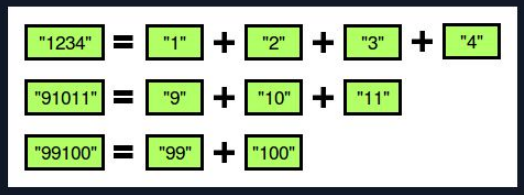
Exit Full Screen View

Problem | Submissions | Leaderboard | Discussions | Editorial

A numeric string, s, is beautiful if it can be split into a sequence of two or more positive integers, $a[1], a[2], \ldots, a[n]$, satisfying the following conditions:

1. $a[i] - a[i-1] = 1$ for any $1 < i \leq n$ (i.e., each element in the sequence is 1 more than the previous element).
2. No $a[i]$ contains a leading zero. For example, we can split $s = 10203$ into the sequence $\{1, 02, 03\}$, but it is not beautiful because 02 and 03 have leading zeroes.
3. The contents of the sequence cannot be rearranged. For example, we can split $s = 312$ into the sequence $\{3, 1, 2\}$, but it is not beautiful because it breaks our first constraint (i.e., $1 - 3 \neq 1$).

The diagram below depicts some beautiful strings:



Perform q queries where each query consists of some integer string s. For each query, print whether or not the string is beautiful on a new line. If it is beautiful, print `YES x`, where x is the first number of the increasing sequence. If there are multiple such values of x, choose the smallest. Otherwise, print `NO`.

Function Description

Complete the separateNumbers function in the editor below.

separateNumbers has the following parameter:

- s: an integer value represented as a string

Prints

- string: Print a string as described above. Return nothing.

Input Format

The first line contains an integer q, the number of strings to evaluate.

Each of the next q lines contains an integer string s to query.

Constraints

$1 \leq q \leq 10$

Change Theme | Language: Java 7

```java
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    private static boolean check(final String s, final long n) {
        final String numberString = String.valueOf(n);
        if (s.length() < numberString.length()) {
            return false;
        } else if (s.startsWith(numberString)) {
            final String nextString = s.substring(numberString.length());
            final long nextNumber = n + 1L;
            return nextString.isEmpty() || check(nextString, nextNumber);
        } else {
            return false;
        }
    }

    private static long breakString(final String s) {
        for (int i = 0; i < s.length() / 2; i++) {
            final long start = Long.parseLong(s.substring(0, i + 1));
            final String nextString = s.substring(i + 1);
            final long nextNumber = start + 1L;
            if (check(nextString, nextNumber)) {
                return start;
            }
        }
        return -1L;
    }

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int q = in.nextInt();
        for(int a0 = 0; a0 < q; a0++){
```

Line: 47 Col: 2

Upload Code as File | Test against custom input | Run Code | Submit Code

30°C Haze | Search | ENG IN | 8:54 AM 7/12/2023