TELUS AI - DSA - Dec 2024

Program 1 -

You are given an integer array nums sorted in ascending order (with distinct values), but rotated at
an unknown pivot. Find the minimum element in the array.
The solution must run in O(log n) time complexity.

Example 1:
Input: nums = [3, 4, 5, 1, 2]
Output: 1
Explanation: The array is rotated at pivot 3, and the minimum element is 1.

Example 2:
Input: nums = [4, 5, 6, 7, 0, 1, 2]
Output: 0
Explanation: The array is rotated at pivot 4, and the minimum element is 0.

Program 2 -

Given an integer array `nums`, find the contiguous subarray (containing at least one number) which
has the largest sum and return its sum.

Example 1:
Input: nums = [-2,1,-3,4,-1,2,1,-5,4]
Output: 6
Explanation: The subarray [4,-1,2,1] has the largest sum, which is 6.

Example 2:
Input: nums = [1]
Output: 1
Explanation: The array contains only one element, which is the largest sum.

Example 3:
Input: nums = [5,4,-1,7,8]
Output: 23
Explanation: The entire array is the subarray with the largest sum.

Program 3 -

Computer crash

Your computer has two RAM sticks containing memoryl and memory2 bits of memory respectively. A virus has entered your computer. Each second since its entry, it is consuming increasing bits of memory. On second 1, it consumes 1 bit of memory, on second 2 it consumes 2 bits of memory, on second 3 it consumes 3 bits of memory, and so on. It consumes memory from the RAM which contains a higher memory or from the first RAM if both have the same memory left. If neither RAM has the memory needed to feed the virus, the computer crashes.

Find out the first second when the computer crashes and the memory contents of the RAMs at that second.

Note: If the virus cannot get full memory, it does not feed the partial memory.

Function description

Complete the function solution) provided in the editor. The function takes the following 2 parameters and returns the solution.

- memory: Returns the memory of RAM 1
- memory2: Returns the memory of RAM 2

Input format for custom testing

Note: Use this input format if you are testing against custom input or writing code in a language where we don't provide boilerplate code

- The first line contains memory denoting the memory of RAM 1.
- The second line contains memory denoting the memory of RAM 2.

Output format

Print 3 integers, representing the second when the computer crashes and the memory remaining in RAM 1 and RAM 2 respectively.

Explanation

Given Input:

memory1 = 4 , memory2 = 4

Output: 4 0 2

Approach :

At second 1, the virus feeds from RAM 1. The remaining memory is [3,4).

At second 2, the virus feeds from RAM 2. The remaining memory is [3,2).

At second 3, the virus feeds from RAM 1. The remaining memory is (0,2).

The computer crashes at second 4.

```
func solution(memory1: Int, memory2: Int) -> [Int] {
    let ans: [Int] = [1, 2, 3]
    return ans
}
```

Program 4 -

You are given an integer `N`. You need to use Base-62 encoding and return the resulting string. In Base-62 encoding:**

- `0-9` → `0-9`
- `10-35` → `A-Z`
- `36-61` → `a-z`

For each given integer, print the Base-62 encoded string.

Function description

Complete the function solve). This function takes the following parameter and returns the required string:

• N: Represents the integer to be encoded

Input format for custom testing

Note: Use this input format if you are testing against custom input or writing code in a language where we don't provide boilerplate code

• The first line contains T, which represents the number of test cases.
• For each test case:
• The single line contains a positive integer N.

Output format

For each test case in a new line, print the encoded string.

```
func solve(N: Int64) -> String {
    return ""
}
```