ENTRI elevate தமிழ்

accenture CODING QUESTIONS & SOLUTION 2024

Accenture Coding Round Details

PART I

ENTRI
elevate
தமிழ்

Writing the code from scratch the candidate will be given 2 questions and 45 minutes of time to solve the problem.
The m language to write the codes are

1. C
2. C++
3. Java
4. Python
5. Dot Net

Number of questions	2 Questions
Type of Test	Non- Adaptive
Time Duration	45 minutes
Negative Marking	No

ENTRI
elevate
தமிழ்

1. CHOCOLATE DISTRIBUTION

Problem Description:

The function accepts an integer array ‘arr’ of size ‘n’ as its argument. Each element of ‘arr’ represents the number of chocolates distributed to a person. The function needs to return the minimum number of chocolates that need to be distributed to each person so that the difference between the chocolates of any two people is minimized.

Example:

Input:

n: 5

arr: 10 4 12 3 1

Output:

3

ENTRI
elevate
தமிழ்

Solution:

```
def min_chocolates(arr):
    arr.sort()
    min_chocolates_needed = float('inf')

    for i in range(len(arr) - 1):
        chocolates_needed = arr[i + 1] - arr[i]
        min_chocolates_needed = min(min_chocolates_needed, chocolates_needed)

    return min_chocolates_needed

arr1 = [10, 4, 12, 3, 1]
output1 = min_chocolates(arr1)
print(output1)
```

ENTRI elevate
தமிழ்

2. PARKING LOT

Problem Description:

The function accepts a character array ‘arr’ of size ‘n’ as its argument. Each element of ‘arr’ represents the status of a parking slot, where ‘S’ represents an empty slot and ‘X’ represents an occupied slot. The function needs to return the maximum number of cars that can be parked in the parking lot. It is assumed that two cars cannot occupy the same slot and cars can only park in consecutive empty slots.

Example:

Input:

n: 16

arr:
XXXSXXSXXS
SXXSXX

Output:

7

ENTRI
elevate
தமிழ்

Solution:

```
#include <stdio.h>

int max_cars_parked(int n, char arr[]) {
    int max_cars = 0;
    int current_cars = 0;

    for (int i = 0; i < n; i++) {
        if (arr[i] == 'S') {
            current_cars++;
        } else {
            max_cars = (max_cars > current_cars) ? max_cars : current_cars;
            current_cars = 0;
        }
    }

    max_cars = (max_cars > current_cars) ? max_cars : current_cars;
    return max_cars;
}

int main() {
    int n = 16;
    char arr[] = "XXXSXXSXXSSXXSXX";
    int result = max_cars_parked(n, arr);
    printf("%d\n", result);

    return 0;
}
```

ENTRI
elevate
தமிழ்

3. STRING TRANSFORMATION

Problem Description:

The function accepts a string ‘str’ as its argument. The function needs to return the transformed string by replacing all occurrences of the character ‘a’ with the character ‘b’ and vice versa.

Example:

Input:

str:
abaabbcc

Output:

bbbbaaac

Solution:

```
def transform_string(str):
    transformed_string = ""

    for i in range(len(str)):
        if str[i] == 'a':
            transformed_string += 'b'
        elif str[i] == 'b':
            transformed_string += 'a'
        else:
            transformed_string += str[i]

    return transformed_string

str = "abaabbcc"
transformed_string = transform_string(str)
print(transformed_string)
```

ENTRI elevate
தமிழ்

4. ARRAY EQUILIBRIUM

Problem Description:

The function accepts an integer array ‘arr’ of size ‘n’ as its argument. The function needs to return the index of an equilibrium point in the array, where the sum of elements on the left of the index is equal to the sum of elements on the right of the index. If no equilibrium point exists, the function should return -1.

Example:

Input:

n: 5

arr: 1 3 5 7 3

Output:

3

ENTRI elevate
தமிழ்

Solution:

```
int find_equilibrium_index(int arr[], int n) {
    int prefix_sums[n + 1];

    for (int i = 0; i < n; i++) {
        prefix_sums[i + 1] = prefix_sums[i] + arr[i];
    }

    for (int i = 0; i < n; i++) {
        int left_sum = prefix_sums[i];
        int right_sum = prefix_sums[n] - prefix_sums[i + 1];

        if (left_sum == right_sum) {
            return i;
        }
    }

    return -1;
}

int main() {
    int arr[] = {1, 3, 5, 7, 3};
    int n = sizeof(arr) / sizeof(arr[0]);

    int equilibrium_index = find_equilibrium_index(arr, n);

    if (equilibrium_index == -1) {
        printf("No equilibrium index found\n");
    } else {
        printf("Equilibrium index: %d\n", equilibrium_index);
    }

    return 0;
}
```

ENTRI elevate
தமிழ்

5. ARRAY ROTATION

Problem Description:

The function accepts an integer array ‘arr’ of size ‘n’ and an integer ‘d’ as its argument. The function needs to rotate the array ‘arr’ by ‘d’ positions to the right. The rotation should be done in place, without using any additional memory.

Example:

Input:

n: 5

arr: 1 2 3 4 5

d: 3

Output:

3 4 5 1 2

Solution:

```
def rotate_array(arr, n, d):
    d = d % n

    for i in range(d):
        temp = arr[0]

        for j in range(n - 1):
            arr[j] = arr[j + 1]

        arr[n - 1] = temp
```