Accenture Coding Questions and Solution 2024

Accenture Coding Round Details

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

- C
- C++
- Java
- Python
- Dot Net

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

Accenture Coding Questions And Answers 2024

1. Chocolate Distribution

(Asked in Accenture OnCampus 10 Aug 2022, Slot 2)

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

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)
```

2. **Parking Lot (Asked in Accenture OnCampus 10 Aug 2022, Slot 3)**

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: XXXSXXSXXSSXXSXX
Output:
7

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;
}
```

3. **String Transformation**

(Asked in Accenture OnCampus 10 Aug 2022, Slot 4)

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)
```

4. **Array Equilibrium**
(Asked in Accenture OnCampus 10 Aug 2022, Slot 5)

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

Solution :

```
#include <stdio.h>
```

```
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;
}
```

5. Array Rotation
(Asked in Accenture OnCampus 10 Aug 2022, Slot 6)

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
```

6. **Substring Search**
(Asked in Accenture OnCampus 10 Aug 2022, Slot 7)

Problem Description:

The function accepts two strings ‘str1’ and ‘str2’ as its argument. The function needs to return the index of the first occurrence of substring ‘str2’ in string ‘str1’ or -1 if the substring is not found.

Example:

Input:
str1: "Hello, World!"
str2: "World"
Output:
7

Solution :

```
def find_substring(str1, str2):
    n = len(str1)
    m = len(str2)
```

```
    for i in range(n - m + 1):
        if str1[i:i + m] == str2:
            return i

    return -1
```

7. **Palindrome Check**
(Asked in Accenture OnCampus 10 Aug 2022, Slot 8)

Problem Description:

The function accepts a string ‘str’ as its argument. The function needs to determine whether the string is a palindrome or not. A palindrome is a word or phrase that reads the same backward as forward.

Example:

Input:
str: "madam"
Output:
1

Solution:
```
def is_palindrome(str):
    n = len(str)

    for i in range(n // 2):
        if str[i] != str[n - 1 - i]:
            return False

    return True
```

8. **Reverse Words**
(Asked in Accenture OnCampus 10 Aug 2022, Slot 9)

Problem Description:

The function accepts a string ‘str’ as its argument. The function needs to reverse the order of the words in the string.

Example:

Input:

str: "Hello, World!"
Output:
!dlroW ,olleH

Solution:

```
def reverse_words(str):
    words = str.split(' ')
    reversed_words = words[::-1]
    return ' '.join(reversed_words)
```

9. Find Two Numbers with Sum N

Problem Description:

Given an array of integers and an integer sum, find a pair of numbers (a, b) in the array where a + b = sum.

Input:

An array of integers
An integer sum
Output:

An array of two integers representing the pair (a, b) or -1 if no such pair exists

Solution :

```
def find_two_numbers(arr, sum):
    seen = set()

    for num in arr:
        target = sum - num

        if target in seen:
            return [num, target]

        seen.add(num)

    return -1
```

Explanation:

Given an array of integers, such as [5, 2, 4, 1, 3], and an integer sum, such as 9, the algorithm should determine that the pair (a, b) = (2, 7) or (4, 5) satisfies the condition a + b = sum. If no such pair exists, the algorithm should return -1.

10. Maximum Subarray Sum

Problem Description:
Given an array of integers, find the maximum subarray sum. A subarray is a contiguous subsequence of the array.

Input:

An array of integers
Output:

An integer representing the maximum subarray sum

Solution :

```
def max_subarray_sum(arr):
    max_sum = float('-inf')
    current_sum = 0

    for num in arr:
        current_sum += num

        if current_sum > max_sum:
            max_sum = current_sum

        if current_sum < 0:
            current_sum = 0

    return max_sum
```

Explanation:
Given an array of integers, such as [-2, 1, -3, 4, -1, 2, 1, -5, 4], the algorithm should determine that the maximum subarray sum is 6 ([4, -1, 2, 1]).

11. Character Replacement

Problem Description:

Given a string str, a character ch1, and a character ch2, replace all occurrences of ch1 in str with ch2 and vice versa.

Input:

A string str
A character ch1
A character ch2

Output:

The modified string str where all occurrences of ch1 are replaced with ch2 and vice versa

Example:

Input: str = "apples", ch1 = 'a', ch2 = 'p'
Output: str = "paales"

Solution :

```
def replace_characters(str, ch1, ch2):
    new_str = ''

    for char in str:
        if char == ch1:
            new_str += ch2
        else:
            new_str += char

    return new_str
```

12. Find the Minimum Value and Its Index in the Array

Problem Description:
Given an integer array, find the minimum value and its index in the array.

Input:

An integer array
Output:

The minimum value and its index, separated by a newline character
Example:

Input: [5, 2, 4, 1, 3]
Output: 1 3

Solution :

```
def find_min_and_index(arr):
    min_value = arr[0]
    min_index = 0

    for i in range(1, len(arr)):
        if arr[i] < min_value:
            min_value = arr[i]
            min_index = i

    return f"{min_value} {min_index}”
```

13. Find the Average of All Positive Numbers in an Array

Problem Description:
Given an array of integers, find the average of all positive numbers in the array.

Input:

An integer array
Output:

The average of all positive numbers, or -1 if there are no positive numbers

Example:

Input: [5, 2, -4, 1, 3]
Output: 3

Solution :

```
def find_average_positives(arr):
    total_positives = 0
    count_positives = 0

    for num in arr:
        if num > 0:
            total_positives += num
```

```
            count_positives += 1

    if count_positives == 0:
        return -1

    return total_positives / count_positives
```

14. Count the Occurrences of a Given Element in an Array

Problem Description:
Given an integer array and an integer element, count the number of occurrences of the element in the array.

Input:

An integer array
An integer element

Output:

The number of occurrences of the element
Example:

Input: [5, 2, 4, 1, 2], 2
Output: 2

Solution :

```
public int countOccurrences(int[] arr, int element) {
  int count = 0;
  for (int num : arr) {
    if (num == element) {
      count++;
    }
  }
  return count;
}
```

15. Check if an Array Contains a Given Element

Problem Description:

Given an integer array and an integer element, check if the array contains the element.

Input:

An integer array
An integer element

Output:

True if the element is found, False otherwise

Example:

Input: [5, 2, 4, 1, 3], 2
Output: True

Solution :

```
public boolean containsElement(int[] arr, int element) {
  for (int num : arr) {
    if (num == element) {
      return true;
    }
  }
  return false;
}
```

16. **Problem Statement: Calculate Prime Sum**

Implement the function:

Int CalculatePrimeSum(int m, int n);

Calculate and return the sum of prime numbers between 'm' and 'n' (inclusive).
Note: 0 < m <= n

Example:

Input:
m : 10
n : 50

Output:
158

Solution :

```
public int calculatePrimeSum(int m, int n) {
  int sum = 0;
  for (int i = m; i <= n; i++) {
    if (isPrime(i)) {
      sum += i;
    }
  }
  return sum;
}

private boolean isPrime(int number) {
  if (number <= 1) {
    return false;
  }
  for (int i = 2; i <= Math.sqrt(number); i++) {
    if (number % i == 0) {
      return false;
    }
  }
  return true;
}
```

17. Problem Statement: Digit Sum Difference

Implement the function:

Int DigitSumDifference(int m, int n);

Calculate and return the absolute difference between the sum of digits of numbers divisible by 4 and the sum of digits of numbers divisible by 7, in the range from 'm' to 'n' (inclusive).
Note: 0 < m <= n

Example:

Input:
m : 50
n : 120

Output:
2

Solution :

```
public int digitSumDifference(int m, int n) {
  int sumDivisibleBy4 = 0;
  int sumDivisibleBy7 = 0;
  for (int i = m; i <= n; i++) {
    int sum = getDigitSum(i);
    if (i % 4 == 0) {
      sumDivisibleBy4 += sum;
    } else if (i % 7 == 0) {
      sumDivisibleBy7 += sum;
    }
  }
  return Math.abs(sumDivisibleBy4 - sumDivisibleBy7);
}

private int getDigitSum(int number) {
  int sum = 0;
  while (number > 0) {
    sum += number % 10;
    number /= 10;
  }
  return sum;
}
```

18. **Problem Statement: Fibonacci Sum**

Implement the function:

Int FibonacciSum(int m, int n);
Calculate and return the sum of Fibonacci numbers in the range from 'm' to 'n' (inclusive).
Note: 0 < m <= n

Example:

Input:
m : 5
n : 20

Output:
52

Solution :

```
 public int fibonacciSum(int m, int n) {
  if (m == 0) {
    return 0;
  }
  int[] fibonacciNumbers = new int[n + 1];
  fibonacciNumbers[0] = 0;
  fibonacciNumbers[1] = 1;
  int sum = 0;
  for (int i = 2; i <= n; i++) {
    fibonacciNumbers[i] = fibonacciNumbers[i - 1] + fibonacciNumbers[i - 2];
    if (i >= m) {
      sum += fibonacciNumbers[i];
    }
  }
  return sum;
}
```

19. Problem Statement: Reverse and Add

Implement the function:

Int ReverseAndAdd(int m, int n);

Calculate and return the sum of numbers obtained by reversing the digits of each number in the range from 'm' to 'n' (inclusive).
Note: 0 < m <= n

Example:

Input:
m : 21
n : 35

Output:
288

Solution :

```
def reverseAndAdd(m, n):
    sum = 0
    for i in range(m, n + 1):
        reversedNum = 0
```

```
        while i > 0:
            reversedNum = reversedNum * 10 + i % 10
            i //= 10
        sum += reversedNum + i
    return sum

result = reverseAndAdd(21, 35)
print(result)
```

20. **Problem Statement: Square Root Difference**

Implement the function:

Int SquareRootDifference(int m, int n);
Calculate and return the difference between the sum of square roots of even numbers and the sum of square roots of odd numbers in the range from 'm' to 'n' (inclusive).
Note: 0 < m <= n

Example:

Input:
m : 1
n : 10

Output:
2.29416

Solution :

```
def squareRootDifference(m, n):
    evenSum = 0
    oddSum = 0
    for i in range(m, n + 1):
        if i % 2 == 0:
            evenSum += math.sqrt(i)
        else:
            oddSum += math.sqrt(i)
    return evenSum - oddSum

# Example usage
```

```
result = squareRootDifference(1, 10)
print(result)
```