ourdiu.com | Shared by Dip Acharjee

Daffodil International University

Faculty of Science & Information Technology
Department of Computer Science & Engineering
Final Examination, Summer 2026
Course Code: CSE113, Course Title: Programming and Problem Solving
Level: 1 Term: 2 Batch: 71

Time: 02:00 Hrs **Marks: 40**

Answer ALL Questions

[The figures in the right margin indicate the full marks and corresponding course outcomes. All portions of each question must be answered sequentially.]

1. Identify the errors in the following code? Explain the why they are erroneous. **Marks** | **CO2**

a) **[03]**

```
01  #include <stdio.h>
02  #include <string.h>
03  struct Student {
04      char name[30];
05      char branch[20];
06  };
07  void display(Struct Student s) {
08      printf("Name: %s\n", s.name);
09      printf("Branch: %s\n", s.branch);
10  }
11  void compareName(char a, char b[]) {
12      if(strcmp(ab)==0)
13          printf("Names are same\n");
14      else
15          printf("Names are different\n");
16  }
17  int main() {
18      struct Student s;
19      strcpy(s.name, "Rahul");
20      strcpy(s.branch, "CSE");
21      display('s');
22      printf("Name length: %d\n", strLen(s.name));
23      compareName(s.name, "Rahul");
24      strcat(s.name, ' Kumar');
25      printf("Updated Name: %s\n", s.name);
26      return 0;
27  }
```

b) Rewrite the code without any errors. **[03]**

Generate the output of given codes as follows (write only the output segment in a box):

c) **[03]**

```
#include <stdio.h>
int add(int *x, int *y){
    *x = *x + *y;
    return *x;
}
```

ourdiu.com | Shared by Dip Acharjee

```
void string(char a[]){
   a[1] = 'l';
   printf("%s\n", a);
}
int main(){
   int a = 4, b = 7;
   add(&b, &a);
   printf("The result is %d\n", b);
   char s[10] = "diucse", t[10]="acmlab";
   string(s);
   string(t);
   return 0;
}
```

d) [03]

```
#include <stdio.h>
void pattern(int n, char c){
   if(n == 0)
      return;
   printf("%c%d\n", c, n);
   pattern(n - 1, c + 1);
   printf("%c%d\n", c, n);
}
int main(){
   pattern(3, 'A');
   return 0;
}
```

e) [03]

```
#include <stdio.h>
int sum(int a[], int n){
   if(n == 0)
      return 0;
   int result = a[n-1] + sum(a, n-1);
   printf("%d + previous = %d\n", a[n-1], result);
   return result;
}
int main(){
   int arr[] = {2, 4, 6};
   printf("Total = %d\n", sum(arr, 3));
   return 0;
}
```

2 Identify the problem scenarios given below and write a complete C program for each problem statement. CO3

a) [05]

Write a C program that uses a recursive function to calculate the sum of all elements in a given integer array. The final total must be displayed inside the *main* function.

Input:

- The first line contains an integer n, the number of elements.
- The second line contains n space-separated integers.

Output:

- Print a single integer representing the total sum of the array elements.

Sample Input	Sample Output
6 1 2 3 4 5 6	21

ourdiu.com | Shared by Dip Acharjee

b) Write a C program to identify the boundary elements of matrix and compute the sum of those elements. [05]

Input:

- The first line contains two integers **r** and **c**.
- Next **r** lines each contain **c** integers.

Output:

- Print the sum of the boundary elements.

Sample Input	Sample Output
3 4 1 2 3 4 5 6 7 8 9 10 11 12	Boundary sum = 65

c) You are given a string consisting of lowercase English letters. Write a C program to identify and display the first non-repeating character in the string. If the string does not contain any non-repeating character, display **"No unique character"**. [05]

Input:

- A single string S consisting of **lowercase English letters (a–z) only.**
- The string contains **no spaces.**

Output:

- If the string contains a non-repeating character, print:
  First non-repeating character: <character>
- Otherwise, print: No unique character

Sample Input	Sample Output
swiss	First non-repeating character: w

d) Implement a C program to determine whether a given string is a palindrome or not palindrome. [**Note:** A **palindrome** is a string that reads the same forward and backward. For example, **level**, and **racecar** are palindromes.] [05]

Input:

- A single string S consisting of **lowercase English letters (a–z) only.**
- The string contains **no spaces.**

Output:

- If the string is a palindrome, then print "**Palindrome**" otherwise "**Not Palindrome**"

Sample Input	Sample Output
diuacm	Not Palindrome

e) Write a C program to process a collection of employee records, calculate the average salary of all employees, identify and display the employees whose salaries are higher than the calculated average. [05]

Input:

- First line contains integer **n**.
- Next **n** lines contain: ID <> Name <> Salary

Output:

- Print the average salary.
- Print the names of employees earning above the average.

Sample Input	Sample Output
4 101 John 42000 102 Emma 50000 103 David 39000 104 Sophia 61000	Average Salary = 48000.00 Employees earning above average: Emma Sophia