

# Internal Assessment (Assignment)

**Name:** Deepankar Sharma

**Student Id:** 233512013

**Course Code:** OMC103

**Course Title:** Programming and Problem-Solving

**Assignment No.:** 2

## Part A (10 × 1 = 10 Marks)

MCQ No.	Question
1	The size of a structure is-----
Answer Choices:	a. 1 byte <input checked="" type="checkbox"/> b. Total bytes of all structure members c. 4 bytes d. 2 bytes
2	Structure in C Programming is-----
Answer Choices:	a. Collection of elements of the same data type <input checked="" type="checkbox"/> b. Collection of elements of the different data types c. Set of values d. Built-in data type
3	The maximum number of dimensions in an array is----
Answer Choices:	a. 1 b. 2 c. 3 <input checked="" type="checkbox"/> d. No limit
4	When an array is passed to a function, it is interpreted as ----
Answer Choices:	a. Address of an array <input checked="" type="checkbox"/> b. Address of First Element c. Values of First Element d. Number of elements in an array
5	What is the value of the x in the following statement? X = strcmp(string1, string2);
Answer Choices:	<input checked="" type="checkbox"/> a. 1 b. False <input checked="" type="checkbox"/> c. 0 d. String1 <i>depending upon value of string 1 &amp; string 2</i>
6	int a[5] = {1,2,3,4,5}, what is the value of a[7]?
Answer Keys:	a) 0 <input checked="" type="checkbox"/> b) Garbage Value c) 5 d) 1
7	The format specifier used for printing a string is -----

Answer Keys:	a) %c b) %d <del>c) %s</del> d) %f
8	If a file opening is failed, then fopen will return----
Answer Keys:	<del>a) null (NULL)</del> b) eof() c) Depends on Compiler d) zero
9	The fastest loop in C programming is -----.
Answer Keys:	a) while b) do-while c) for <del>d) All the options</del>
10	What will be the output of the following program? <pre>#include &lt;stdio.h&gt; int main(){     FILE *fp;     char *str;     fp=fopen("demo.txt","r");// demo.txt //:First Semester MCA     while(fgets(str,6,fp)!=NULL)         puts(str);     fclose(fp);     return 0; }</pre>
Answer Key	a) First Semester MCA <del>b) First S</del> c) First Semester d) First Se

## Part B

(5 × 4 = 20 Marks)

Q No.	Question
2	<p>Illustrate different file-handling modes using a suitable C program.</p> <pre>#include &lt;stdio.h&gt;  int main() {     FILE *file;     // Open file in write mode (w)     file = fopen("example.txt", "w");     if (file == NULL) {         printf("Error opening file.\n");         return 1;     }     fprintf(file, "Hello, World!\n");     fclose(file);     // Open file in read mode (r)     file = fopen("example.txt", "r");</pre>

```

    if (file == NULL) {
        printf("Error opening file.\n");
        return 1;
    }
    char line[100];
    while (fgets(line, sizeof(line), file)) {
        printf("%s", line);
    }
    fclose(file);

    // Open file in append mode (a)
    file = fopen("example.txt", "a");
    if (file == NULL) {
        printf("Error opening file.\n");
        return 1;
    }
    fprintf(file, "Appending this line.\n");
    fclose(file);
    return 0;
}

```

The screenshot shows a VS Code editor with a C program in `b2_filemodes.c`. The program reads a file in append mode and prints its contents. The terminal output shows the execution of the program, which successfully appends the text "Hello, World!" to `example.txt`.

```

PS C:\Deepankar\VMCA-semester01\Assignments\Programming and Problem Solving\Assignment 02\codes> cd "c:\Deepankar\VMCA-semester01\Assignments\Programming and Problem Solving\Assignment 02\codes" ; if ($?) { gcc b3_file.c -o b3_file } ; if ($?) { .\b3_file }
Number of Alphabets: 84
Number of Digits: 1
Number of Spaces: 20
PS C:\Deepankar\VMCA-semester01\Assignments\Programming and Problem Solving\Assignment 02\codes> cd "c:\Deepankar\VMCA-semester01\Assignments\Programming and Problem Solving\Assignment 02\codes" ; if ($?) { gcc b2_filemodes.c -o b2_filemodes } ; if ($?) { .\b2_filemodes }
Hello, World!
PS C:\Deepankar\VMCA-semester01\Assignments\Programming and Problem Solving\Assignment 02\codes>

```

Design and develop a C program to read a text and count the number of alphabets, spaces, and digits.

3

```

#include <stdio.h>

int main() {
    FILE *file;
    char ch;
    int digits = 0, spaces = 0, alphabets = 0;

    file = fopen("b3_file.txt", "r");
    if (file == NULL) {

```

```

printf("Error opening file.\n");
return 1;
}

while ((ch = fgetc(file)) != EOF) {
    if(ch >= '0' && ch <= '9')
        digits++;
    else if(ch == ' ')
        spaces++;
    else if((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z'))
        alphabets++;
}

printf("Number of Alphabets: %d\n", alphabets);
printf("Number of Digits: %d\n", digits);
printf("Number of Spaces: %d\n", spaces);

fclose(file);
return 0;
}

```

```

#include <stdio.h>

int main() {
    FILE *file;
    char ch;
    int digits = 0, spaces = 0, alphabets = 0;

    file = fopen("b3_file.txt", "r");
    if (file == NULL) {
        printf("Error opening file.\n");
        return 1;
    }

    while ((ch = fgetc(file)) != EOF) {
        if(ch >= '0' && ch <= '9')
            digits++;
        else if(ch == ' ')
            spaces++;
        else if((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z'))
            alphabets++;
    }

    printf("Number of Alphabets: %d\n", alphabets);
    printf("Number of Digits: %d\n", digits);
    printf("Number of Spaces: %d\n", spaces);

    fclose(file);
    return 0;
}

```

Terminal Output:

```

PS C:\Deepankar\WCA-semester01\Assignments\Programming and Problem Solving\Assignment 02\codes> cd "c:\Deepankar\WCA-semester01\Assignments\Programming and Problem Solving\Assignment 02\codes\"; if ($?) { gcc b3_file.c -o b3_file }; if ($?) { .\b3_file }
Number of Alphabets: 84
Number of Digits: 1
Number of Spaces: 20
PS C:\Deepankar\WCA-semester01\Assignments\Programming and Problem Solving\Assignment 02\codes>

```

Write a C program to add two complex numbers using structures. Use a concept of structures for multiple records of complex numbers.

4

```

#include <stdio.h>

typedef struct Complex {
    float real;
    float imag;
} Complex;

Complex add(Complex n1, Complex n2) {
    Complex temp;
    temp.real = n1.real + n2.real;
}

```

```

        temp.imag = n1.imag + n2.imag;
        return temp;
    }

    int main() {
        Complex n1, n2, result;
        printf("For 1st complex number \n");
        printf("Enter real and imaginary part respectively:\n");
        scanf("%f %f", &n1.real, &n1.imag);
        printf("\nFor 2nd complex number \n");
        printf("Enter real and imaginary part respectively:\n");
        scanf("%f %f", &n2.real, &n2.imag);
        result = add(n1, n2);
        printf("Sum = %.1f + %.1fi", result.real, result.imag);
        return 0;
    }

```

The screenshot shows the Visual Studio Code interface with a C program open. The code defines a `Complex` struct and an `add` function. The `main` function prompts the user for two complex numbers and prints their sum. The terminal output shows the program running successfully with input 4 56 and 35 6, resulting in a sum of 39.0 + 62.0i.

Write the advantages and drawbacks of recursion. Write a c-program using the recursive function for Binary to Decimal Conversion.

### Advantages of Recursion:

- ① Simplicity → Recursion makes the code simpler & elegant
- ② Easy to use → Similar function calls are made which makes recursion very easy to use
- ③ Data structures like trees & graphs and also Dynamic programming problems are implemented using recursion.

### Drawbacks of Recursion

- ① Each recursive call increase the stack size overhead.
- ② Recursive calls are expensive in terms of memory & time
- ③ failed Base condition may lead to infinite recursive calls.

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

int binaryToDecimal(int binary, int n) {
    if(binary == 0) {
        return 0;
    } else {
        return ((binary%10) * pow(2,n)) + binaryToDecimal(binary/10, n+1);
    }
}

int main() {
    int binary;
    printf("Enter a binary number: ");
    scanf("%d", &binary);
    printf("Decimal: %d", binaryToDecimal(binary, 0));
    return 0;
}
```

Write a c-program using structures to read, write, compute average - marks and display the students scoring above and below the average marks for a class of N students

7

```
#include <stdio.h>

typedef struct {
    char name[50];
    int marks;
} Student;

int main() {
    int i, n, sum = 0;
```

```

float average;
printf("Enter the number of students: ");
scanf("%d", &n);
Student students[n];
for(i = 0; i < n; i++) {
    printf("Enter name and marks for student %d: ", i+1);
    scanf("%s %d", students[i].name, &students[i].marks);
    sum += students[i].marks;
}
average = (float)sum / n;
printf("\nAverage marks = %.2f\n", average);
printf("\nStudents scoring above the average:\n");
for(i = 0; i < n; i++) {
    if(students[i].marks > average) {
        printf("%s\n", students[i].name);
    }
}
printf("\nStudents scoring below the average:\n");
for(i = 0; i < n; i++) {
    if(students[i].marks < average) {
        printf("%s\n", students[i].name);
    }
}
return 0;
}

```

The screenshot shows the Visual Studio Code interface with the following components:

- Explorer Panel:** Displays the file structure of the project, including source files (e.g., `b2_filemodes.c`, `b7_scoring.c`) and their corresponding executables (e.g., `b7_scoring.exe`).
- Editor Panel:** Shows the source code of `b7_scoring.c`, which includes the C program logic for calculating the average and filtering students based on their marks.
- Terminal Panel:** Displays the output of the program execution. The output shows the user entering the number of students (5) and their names and marks, followed by the calculated average (39.00) and the list of students scoring above and below the average.

**Terminal Output:**

```

PS C:\Deepankar\WCA-semester01\Assignments\Programming and Problem Solving\Assignment 02\codes> cd "c:\Deepankar\WCA-semester01\Assignments\Programming and Problem Solving\Assignment 02\codes" ; if ($?) { gcc b7_scoring.c -o b7_scoring } ; if ($?) { .b7_sc
Enter the number of students: 5
Enter name and marks for student 1: 24
34
Enter name and marks for student 2: 56
45
Enter name and marks for student 3: 67
46
Enter name and marks for student 4: 45
35
Enter name and marks for student 5: 35
35

Average marks = 39.00

Students scoring above the average:
56
67

Students scoring below the average:
24
45
35

```