Solutions to Question Paper

Group-A (Very Short Answer Type Questions)

1. Answer any ten of the following:

(i) What is the use of `continue` in loop?

- `continue` is used to skip the remaining part of the loop iteration and move to the next iteration without exiting the loop.

(ii) What is the string function used to reverse a string?

- The string function to reverse a string is `strrev()` in C.

(iii) What is the time complexity of Bubble sort?

- The time complexity of Bubble Sort is $O(n^2)$, where `n` is the number of elements in the array.

(iv) What are preprocessors?

- Preprocessors are directives in C (such as `#include`, `#define`) that provide instructions to the compiler to preprocess the source code before compilation.

(v) Recursion uses which data structure?

- Recursion uses the **stack data structure.**

(vi) What header file is used for string operations?

- The header file used for string operations is `#include <string.h>`.

(vii) What is nested `if-else`?

- A nested `if-else` refers to an `if` statement inside another `if` or `else` block.

(viii) What is the use of `goto` statement?

- The `goto` statement is used to transfer control to a labeled statement within the same function. It is generally not recommended as it can make the code hard to follow.

(ix) What is linear searching in an array?

- Linear searching is a method of finding an element in an array by checking each element sequentially until the desired element is found.

(x) What is a Macro?

- A macro is a fragment of code which has been given a name. Whenever the name is used, it is replaced by the contents of the macro. Macros are defined using the `#define` directive.

(xi) What is the value of the following if `a = 2`?

printf("%d", --a);

- The value of `a` will be decremented before printing, so the output will be `1`.

(xii) What is `goto` statement in C?

- The `goto` statement allows the program to jump to another location in the code, typically labeled by a label name. However, it is generally discouraged as it can make the flow of the program difficult to understand.

Group-B (Short Answer Type Questions)

2. What is a structure? What is an array of structures? Show with an example.

- Structure: A structure is a user-defined data type that allows grouping of variables of different data types under a single name.

- Array of structures: An array of structures is a collection of variables of structure type that are stored in contiguous memory locations.

Example:

```
struct Student {
    char name[50];
    int roll;
    float marks;
};
struct Student s[10];  // Array of structures
```

3. Write a program to find the roots of a quadratic equation.

```
#include <stdio.h>
#include <math.h>
int main() {
    float a, b, c, discriminant, root1, root2;
    printf("Enter coefficients a, b, and c: ");
    scanf("%f %f %f", &a, &b, &c);
    discriminant = b*b - 4*a*c;
    if (discriminant > 0) {
        root1 = (-b + sqrt(discriminant)) / (2*a);
        root2 = (-b - sqrt(discriminant)) / (2*a);
        printf("Roots are: %.2f and %.2f\n", root1, root2);
```

```
    } else if (discriminant == 0) {
        root1 = -b / (2*a);
        printf("Root is: %.2f\n", root1);
    } else {
        printf("Roots are complex and imaginary.\n");
    }
    return 0;
}
```

4. Draw flowchart to evaluate the factorial of a number.

- Start -> Read number `n` -> Initialize `fact = 1` -> While `n > 0` -> `fact = fact * n` -> Decrease `n` by 1 -> If `n = 0`, print `fact` -> End

5. What is a conditional operator?

- A conditional operator is a ternary operator that takes three operands and evaluates an expression based on a condition. Syntax:

condition ? expression1 : expression2;

If the condition is true, `expression1` is evaluated; otherwise, `expression2` is evaluated.

6. What will be the output of the following code?

```
#include <stdio.h>
int main() {
    int a[5] = {1, 2, 3, 4, 5};
    int i;
    for (i = 0; i < 5; i++)
        if ((char)a[i] == '5')
```

```
            printf("%d\n", a[i]);
        else
            printf("FAIL\n");
    return 0;
}
```

- The output will be:

  FAIL

  FAIL

  FAIL

  FAIL

  FAIL

Programming Questions and Solutions

7. (a) Difference Between Array and Structure:

Array: An array is a collection of elements of the same data type stored at contiguous memory locations.

Structure: A structure is a user-defined data type that can hold elements of different data types.

Example:

```
// Array Example
int arr[5] = {1, 2, 3, 4, 5};

// Structure Example
struct Student {
    char name[50];
    int roll;
    float marks;
};
struct Student s1;
```

7. (b) Differences Between Structure and Union:

Structure: In a structure, each member has its own memory location, and all members can be accessed and used simultaneously.

Union: In a union, all members share the same memory location, and only one member can be accessed at a time.

Example:

```
// Structure Example
struct Data {
    int i;
    float f;
    char str[20];
} data1;

// Union Example
union Data {
    int i;
    float f;
    char str[20];
} data2;
```

7. (c) Program Using Structure to Take 10 Students' Information and Print Records with Average Marks:

```
#include <stdio.h>

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

int main() {
    struct Student students[10];
```

```
    float total_marks = 0.0, average;

    for(int i = 0; i < 10; i++) {
        printf("Enter name, roll, and marks for student %d:\n", i+1);
        scanf("%s %d %f", students[i].name, &students[i].roll, &students[i].marks);
        total_marks += students[i].marks;
    }

    average = total_marks / 10;

    printf("\nStudent Details:\n");
    for(int i = 0; i < 10; i++) {
        printf("Name: %s, Roll: %d, Marks: %.2f\n", students[i].name, students[i].roll, students[i].marks);
    }
    printf("\nAverage Marks: %.2f\n", average);

    return 0;
}
```

8. (a) Program to Check if a Number is Even or Odd:

```
#include <stdio.h>

int main() {
    int num;
    printf("Enter an integer: ");
    scanf("%d", &num);
```

```
    if (num % 2 == 0)
        printf("%d is even.\n", num);
    else
        printf("%d is odd.\n", num);

    return 0;
}
```

8. (b) Program to Check if a Number is Prime or Not:

```
#include <stdio.h>

int main() {
    int num, i, isPrime = 1;

    printf("Enter an integer: ");
    scanf("%d", &num);

    if (num <= 1) isPrime = 0;

    for (i = 2; i <= num / 2; ++i) {
        if (num % i == 0) {
            isPrime = 0;
            break;
        }
    }
```

```
    if (isPrime)
        printf("%d is a prime number.\n", num);
    else
        printf("%d is not a prime number.\n", num);

    return 0;
}
```

8. (c) Program to Add All Even Numbers from an Array:

```
#include <stdio.h>

int main() {
    int arr[10], i, sum = 0;

    printf("Enter 10 integers:\n");
    for(i = 0; i < 10; i++) {
        scanf("%d", &arr[i]);
    }

    for(i = 0; i < 10; i++) {
        if(arr[i] % 2 == 0)
            sum += arr[i];
    }

    printf("Sum of all even numbers: %d\n", sum);
```

```
    return 0;

}
```

Question 9: Theory-Based Questions

(a) What is an Operating System?
An Operating System (OS) is system software that manages computer hardware, software resources, and provides common services for computer programs. It acts as an intermediary between users and the computer hardware.

(b) What are the functions of an Operating System?
The main functions of an OS include:

- **Resource Management: Manages hardware resources such as CPU, memory, and I/O devices.**
- **User Interface: Provides a user interface, such as a command-line interface (CLI) or graphical user interface (GUI).**
- **File Management: Handles files on various storage devices, including creating, deleting, and organizing files.**
- **Security and Access Control: Ensures authorized access to system resources and data.**
- **Process Management: Manages processes, including multitasking, process synchronization, and inter-process communication.**
- **Device Management: Controls and manages hardware devices through device drivers.**

(c) What is Booting?
Booting is the process of starting a computer and loading the operating system into the computer's main memory (RAM) so that it can begin executing tasks. The booting process includes power-on self-test (POST), loading the bootloader, and initializing the operating system.

(d) What are the differences between a Compiler and an Interpreter?

- **Compiler: Translates the entire source code of a program into machine code at once, which is then executed by the computer. The compiled code is typically saved as an executable file.**
- **Interpreter: Translates source code line-by-line into machine code and executes it immediately. Unlike a compiler, it does not produce an intermediate executable file.**

Question 10: Programming Tasks

(a) Write a program to print the diagonal matrix from a 5 x 5 matrix.

Here is a C program that prints the diagonal elements of a 5x5 matrix:

c

Copy code

```
#include <stdio.h>


int main() {
```

```
    int matrix[5][5] = {
        {1, 2, 3, 4, 5},
        {6, 7, 8, 9, 10},
        {11, 12, 13, 14, 15},
        {16, 17, 18, 19, 20},
        {21, 22, 23, 24, 25}
    };

    printf("Diagonal elements of the matrix are:\\n");
    for (int i = 0; i < 5; i++) {
        printf("%d ", matrix[i][i]);
    }

    return 0;
}
```

This program initializes a 5x5 matrix with values from 1 to 25 and prints the diagonal elements (1, 7, 13, 19, 25).

(b) Write a program to concatenate two strings without using the strcat() function in C.

Here is a C program to manually concatenate two strings:

c

Copy code

```
#include <stdio.h>

void concatenate(char result[], char str1[], char str2[]) {
    int i = 0, j = 0;

    // Copy str1 to result
    while (str1[i] != '\0') {
        result[i] = str1[i];
        i++;
    }
```

```
    // Copy str2 to result
    while (str2[j] != '\0') {
        result[i] = str2[j];
        i++;
        j++;
    }

    result[i] = '\0'; // Null-terminate the concatenated string
}

int main() {
    char str1[] = "Hello";
    char str2[] = "World";
    char result[100];

    concatenate(result, str1, str2);

    printf("Concatenated string: %s\\n", result);

    return 0;
}
```

This program manually concatenates two strings by copying characters from both strings into a result array.

(c) Write a program to sort n elements from an array.

Here is a simple C program to sort an array of n elements using the bubble sort algorithm:

c

Copy code

```
#include <stdio.h>

void bubbleSort(int arr[], int n) {
```

```
    int i, j, temp;
    for (i = 0; i < n-1; i++) {
        for (j = 0; j < n-i-1; j++) {
            if (arr[j] > arr[j+1]) {
                temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
        }
    }
}

int main() {
    int arr[] = {64, 34, 25, 12, 22, 11, 90};
    int n = sizeof(arr)/sizeof(arr[0]);
    int i;

    bubbleSort(arr, n);

    printf("Sorted array: \\n");
    for (i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }

    return 0;
}
```

This program sorts an array of integers using the bubble sort algorithm and prints the sorted array.

Question 11: Number Systems and Operations

(a) Convert Binary to Decimal of $(1101.01)_2$
To convert the binary number 1101.01 to decimal:

Steps:

1. Convert the integer part 1101_2 to decimal:
   - 1101_2 represents:
     $1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
   - Calculation:
     $$1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = 8 + 4 + 0 + 1 = 13$$

   So, $1101_2 = 13_{10}$.
2. Convert the fractional part 0.01_2 to decimal:
   - 0.01_2 represents:
     $0 \times 2^{-1} + 1 \times 2^{-2}$
   - Calculation:
     $$0 \times 0.5 + 1 \times 0.25 = 0 + 0.25 = 0.25$$

   So, $0.01_2 = 0.25_{10}$.
3. Combine the integer and fractional parts:
   $$1101.01_2 = 13 + 0.25 = 13.25_{10}$$

Final Answer:

$$1101.01_2 = 13.25_{10}$$

(b) What is 2's Complement? Show by example.
2's complement is a method used to represent negative numbers in binary. To find the 2's complement of a binary number:

1. **Invert all the bits (0 becomes 1 and 1 becomes 0).**
2. **Add 1 to the least significant bit (LSB).**

For example, to find the 2's complement of 00101:

1. **Invert the bits: 11010**
2. **Add 1: 11010 + 1 = 11011**

So, the 2's complement of 00101 is 11011.

(c) Using 2's Complement, perform the operation: 10101_2 - 00111_2
To subtract 00111_2 from 10101_2 using 2's complement:

1. Find the 2's complement of 00111_2:
   - Invert the bits: 11000
   - Add 1: 11000 + 1 = 11001_2
2. Add the 2's complement of 00111_2 to 10101_2:
   - $10101_2 + 11001_2 = 101110_2$

The result is 101110_2, which corresponds to 22 in decimal (since the extra bit indicates no overflow).

(d) Convert Decimal to Octal $(5673)_{10}$

To convert the decimal number 5673 to octal:

1. Divide 5673 by 8: $5673 \div 8 = 709$ remainder 1
2. Divide 709 by 8: $709 \div 8 = 88$ remainder 5
3. Divide 88 by 8: $88 \div 8 = 11$ remainder 0
4. Divide 11 by 8: $11 \div 8 = 1$ remainder 3
5. Divide 1 by 8: $1 \div 8 = 0$ remainder 1

So, $5673_{10} = 13051_8$.