MODULE: SE – Fundamentals of Programming

Topics Covered

- **Function**
- **Array**

Que.1 **Write a program to find out the max number from given array using function**

```
#include <stdio.h>

int Max(int arr[], int n) {

    int max_val = arr[0];

    for (int i = 1; i < n; i++) {
        if (arr[i] > max_val) {
            max_val = arr[i];
        }
    }
    return max_val;
}

int main() {
    int array[] = {12, 18, 32, 81, 26, 51};
    int n = sizeof(array) / sizeof(array[0]);

    int max_num = Max(array, n);
    printf("The maximum number in the array is: %d\n", max_num);

    return 0;
}
```

Que.2 WAP of Addition, Subtraction, Multiplication and Division using Switch case.(Must Be Menu Driven)

```
#include <stdio.h>

int main() {
    int choice;
    float num1, num2, result;

    while (1) {

        printf("\nMenu:\n");
        printf("1. Addition\n");
        printf("2. Subtraction\n");
        printf("3. Multiplication\n");
        printf("4. Division\n");
        printf("5. Exit\n");
        printf("Enter your choice (1-5): ");
        scanf("%d", &choice);

        if (choice == 5) {
            printf("Exiting the program.\n");
            break;
        }

        printf("Enter two numbers: ");
        scanf("%f %f", &num1, &num2);

        switch (choice) {
            case 1:
                result = num1 + num2;
                printf("Result of Addition: %.2f\n", result);
                break;
            case 2:
                result = num1 - num2;
                printf("Result of Subtraction: %.2f\n", result);
                break;
            case 3:
                result = num1 * num2;
                printf("Result of Multiplication: %.2f\n", result);
                break;
            case 4:
                if (num2 != 0) {
                    result = num1 / num2;
                    printf("Result of Division: %.2f\n", result);
                } else {
```

```
                printf("Error: Division by zero is not allowed.\n");
            }
            break;
        default:
            printf("Invalid choice. Please select a valid option.\n");
            break;
    }
}

return 0;
}
```

Que.3 **WAP to find reverse of string using recursion**

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

void revString(char str[], int start, int end) {
    if (start >= end) {
        return;
    }
    char temp = str[start];
    str[start] = str[end];
    str[end] = temp;

    revString(str, start + 1, end - 1);
}

int main() {
    char str[100];

    printf("Enter a string: ");
    gets(str);

    int length = strlen(str);

    revString(str, 0, length - 1);

    printf("Reversed string: %s\n", str);

    return 0;
}
```

Que.4 WAP to find factorial using recursion

```
#include <stdio.h>

int factorial(int n) {
    if (n == 0 || n == 1) {
        return 1;
    } else {
        return n * factorial(n - 1);
    }
}

int main() {
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);

    if (num < 0) {
        printf("Factorial is not defined for negative numbers.\n");
    } else {

        int result = factorial(num);

        printf("Factorial of %d is: %d\n", num, result);
    }

    return 0;
}
```

Que.5 WAP to take two Array input from user and sort them in ascending or descending order as per user’s choice

```
#include <stdio.h>

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

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

void print(int arr[], int n) {
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

int main() {
    int n1, n2, ch;

    printf("Enter the  in the first array: ");
    scanf("%d", &n1);
    int arr1[n1];
    printf("Enter the  the first array:\n");
    for (int i = 0; i < n1; i++) {
        scanf("%d", &arr1[i]);
    }

    printf("Enter the  second array: ");
    scanf("%d", &n2);
    int arr2[n2];
    printf("Enter the second array:\n");
    for (int i = 0; i < n2; i++) {
        scanf("%d", &arr2[i]);
    }

    printf("Choose sorting order:\n");
    printf("1. Ascending\n");
    printf("2. Descending\n");
    printf("Enter your choice: ");
    scanf("%d", &ch);

    if (ch == 1) {
        sortAsc(arr1, n1);
        sortAsc(arr2, n2);
        printf("Arrays ascending order:\n");
    } else if (ch == 2) {
        sortDes(arr1, n1);
        sortDes(arr2, n2);
        printf("Arrays descending order:\n");
    } else {
        printf("Invalid choice. Exiting.\n");
        return 1;
    }

    // Print the sorted arrays
    printf("First array: ");
    print(arr1, n1);

    printf("Second array: ");
    print(arr2, n2);

    return 0;
}
```

Que.6 WAP to make addition, Subtraction and multiplication of two matrix using 2-D Array

```
#include <stdio.h>

void inMat(int rows, int cols, int matrix[rows][cols], int
                    matrixNumber) {
    printf("Enter the elements of Matrix %d (%dx%d):\n",
                    matrixNumber, rows, cols);
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            printf("Element [%d][%d]: ", i, j);
            scanf("%d", &matrix[i][j]);
        }
    }
}

void printMat(int rows, int cols, int matrix[rows][cols]) {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            printf("%d\t", matrix[i][j]);
        }
        printf("\n");
    }
}

void add(int rows, int cols, int matrix1[rows][cols], int
                    matrix2[rows][cols], int result[rows][cols]) {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            result[i][j] = matrix1[i][j] + matrix2[i][j];
        }
    }
}

void sub(int rows, int cols, int matrix1[rows][cols], int
                    matrix2[rows][cols], int result[rows][cols]) {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            result[i][j] = matrix1[i][j] - matrix2[i][j];
        }
    }
}
```

```
void multi(int rows1, int cols1, int cols2, int
                    matrix1[rows1][cols1], int matrix2[cols1][cols2], int
                    result[rows1][cols2]) {
    for (int i = 0; i < rows1; i++) {
        for (int j = 0; j < cols2; j++) {
            result[i][j] = 0;
            for (int k = 0; k < cols1; k++) {
                result[i][j] += matrix1[i][k] * matrix2[k][j];
            }
        }
    }
}

int main() {
    int rows, cols;

    printf("Enter the number of rows and columns for Matrix A and
                    Matrix B (same for both): ");
    scanf("%d %d", &rows, &cols);

    int matrixA[rows][cols];
    int matrixB[rows][cols];
    int result[rows][cols];
    int mulResult[rows][cols];


    inMat(rows, cols, matrixA, 1);
    inMat(rows, cols, matrixB, 2);


    add(rows, cols, matrixA, matrixB, result);
    printf("Result of Matrix Addition:\n");
    printMat(rows, cols, result);

    sub(rows, cols, matrixA, matrixB, result);
    printf("Result of Matrix Subtraction:\n");
    printMat(rows, cols, result);

    multi(rows, cols, cols, matrixA, matrixB, mulResult);
    printf("Result of Matrix Multiplication:\n");
    printMat(rows, cols, mulResult);

    return 0;
}
```

Que.7 WAP Find out length of string without using inbuilt function

```
#include <stdio.h>

int len(char str[]) {
    int l = 0;

    while (str[l] != '\0') {
        l++;
    }

    return l;
}

int main() {
    char str[100];

    printf("Enter a string: ");
    gets(str);

    int l = len(str);
    printf("The length of the string is: %d\n", l);

    return 0;
}
```

Que.8 WAP to reverse a string and check that the string is palindrome or not

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

void reverseString(char str[]) {
    int n = strlen(str);
    for (int i = 0; i < n / 2; i++) {
        char temp = str[i];
        str[i] = str[n - i - 1];
        str[n - i - 1] = temp;
    }
}

int isPalindrome(char str[]) {
    int n = strlen(str);
    for (int i = 0; i < n / 2; i++) {
        if (str[i] != str[n - i - 1]) {
            return 0;
        }
    }
    return 1;
}

int main() {
    char str[100], original[100];

    printf("Enter a string: ");
    gets(str);

    strcpy(original, str);


    reverseString(str);
    printf("Reversed string: %s\n", str);

    if (isPalindrome(original)) {
        printf("The string is a palindrome.\n");
    } else {
        printf("The string is not a palindrome.\n");
    }

    return 0;
}
```

Write a program of structure employee that provides the following

a. **information -print and display empno, empname, address andage**

```
#include <stdio.h>

struct Employee {
    int empno;
    char empname[50];
    char address[100];
    int age;
};

void printEmployee(struct Employee emp) {
    printf("Employee Number: %d\n", emp.empno);
    printf("Employee Name: %s\n", emp.empname);
    printf("Address: %s\n", emp.address);
    printf("Age: %d\n", emp.age);
}

int main() {
    struct Employee emp;
    printf("Enter Employee Number: ");
    scanf("%d", &emp.empno);
    getchar();

    printf("Enter Employee Name: ");
    gets(emp.empname);

    printf("Enter Employee Address: ");
    gets(emp.address);

    printf("Enter Employee Age: ");
    scanf("%d", &emp.age);


    printf("\nEmployee Information:\n");
    printEmployee(emp);

    return 0;
}
```

b. **Write a program of structure for five employee that provides the following information -print and display empno, empname, address andage**

```
#include <stdio.h>


struct Employee {
    int empno;
    char empname[50];
    char address[100];
    int age;
};

void printEmployee(struct Employee emp) {
    printf("Employee Number: %d\n", emp.empno);
    printf("Employee Name: %s\n", emp.empname);
    printf("Address: %s\n", emp.address);
    printf("Age: %d\n", emp.age);
    printf("\n");
}
```

```
int main() {
    struct Employee emp[5];


    for (int i = 0; i < 5; i++) {
        printf("Enter details for Employee %d:\n", i + 1);

        printf("Enter Employee Number: ");
        scanf("%d", &emp[i].empno);
        getchar();

        printf("Enter Employee Name: ");
        gets(emp[i].empname);

        printf("Enter Employee Address: ");
        gets(emp[i].address);

        printf("Enter Employee Age: ");
        scanf("%d", &emp[i].age);

        printf("\n");
    }


    printf("Employee Information:\n");
    for (int i = 0; i < 5; i++) {
        printEmployee(emp[i]);
    }

    return 0;
}
```

Que.9 WAP to show difference between Structure and Union.

```
#include <stdio.h>

struct MyStruct {
    int intVal;
    float floatVal;
    char charVal;
};


union MyUnion {
    int intVal;
    float floatVal;
    char charVal;
};

int main() {

    struct MyStruct s;
    s.intVal = 10;
    s.floatVal = 20.5;
    s.charVal = 'A';

    union MyUnion u;
    u.intVal = 10;
    u.floatVal = 20.5;
    u.charVal = 'A';


    printf("Structure values:\n");
    printf("intValue: %d\n", s.intVal);
    printf("floatValue: %.2f\n", s.floatVal);
    printf("charValue: %c\n", s.charVal);

    printf("\nUnion values :\n");
    printf("intValue: %d\n", u.intVal);
    printf("floatValue: %.2f\n", u.floatVal);
    printf("charValue: %c\n", u.charVal);

     printf("\nSize of Structure: %zu bytes\n", sizeof(s));
    printf("Size of Union: %zu bytes\n", sizeof(u));

    return 0;
}
```

Que.10 WAP to perform Palindrome number using for loop and function

```
#include <stdio.h>
int isPali(int num) {
    int actNum = num;
    int revNum = 0;
    int remainder;

    while (num != 0) {
        remainder = num % 10;
        revNum = revNum * 10 + remainder;
        num /= 10;
    }

    if (revNum == actNum) {
        return 1;
    } else {
        return 0;
    }
}

int main() {
    int number;

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

    if (isPali(number)) {
        printf("%d is a palindrome.\n", number);
    } else {
        printf("%d is not a palindrome.\n", number);
    }

    return 0;
}
```

Que.11 WAP to accept 5 numbers from user and display in reverse order using for loop and array

```
#include <stdio.h>

int main() {
    int numbers[5];
    for (int i = 0; i < 5; i++) {
        printf("Enter number %d: ", i + 1);
        scanf("%d", &numbers[i]);
    }

       printf("Numbers in reverse order:\n");
    for (int i = 4; i >= 0; i--) {
        printf("%d\n", numbers[i]);
    }

    return 0;
}
```

Que.12 WAP to accept 5 students name and store it in array

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

int main() {
    char names[5][50];
    for (int i = 0; i < 5; i++) {
        printf("Enter name of student %d: ", i + 1);
        fgets(names[i], 50, stdin);

        names[i][strcspn(names[i], "\n")] = 0;
    }

    printf("\nStored names of students:\n");
    for (int i = 0; i < 5; i++) {
        printf("%s\n", names[i]);
    }

    return 0;
}
```

Que.13 WAP to accept 5 numbers from user and check entered number is even or odd using of array

```
#include <stdio.h>
int main() {
    int numbers[5];
    for (int i = 0; i < 5; i++) {
        printf("Enter number %d: ", i + 1);
        scanf("%d", &numbers[i]);
    }
      for (int i = 0; i < 5; i++) {
        if (numbers[i] % 2 == 0) {
            printf("Enter Number %d (%d) is even.\n", i + 1,
                    numbers[i]);
        } else {
            printf("Enter Number %d (%d) is odd.\n", i + 1,
                    numbers[i]);
        }    }
    return 0;
}
```

Que.14 Perform 2D matrix array

```
#include <stdio.h>
int main() {
    int r, c;
    printf("Enter the number of rows: ");
    scanf("%d", &r);
    printf("Enter the number of columns: ");
    scanf("%d", &c);
    int matrix[r][c];
    printf("Enter the elements of the matrix:\n");
    for (int i = 0; i < r; i++) {
        for (int j = 0; j < c; j++) {
            printf("Element [%d][%d]: ", i + 1, j + 1);
            scanf("%d", &matrix[i][j]);
        }
    }

    printf("\nThe entered matrix is:\n");
    for (int i = 0; i < r; i++) {
        for (int j = 0; j < c; j++) {
            printf("%d\t", matrix[i][j]);
        }
        printf("\n");
    }

    return 0; }
```

Que.15 Store 5 numbers in array and sort it in ascending order

```
#include <stdio.h>

int main() {
   int num[5];
   int temp;

   printf("Enter 5 numbers:\n");
   for (int i = 0; i < 5; i++) {
      printf("Number %d: ", i + 1);
      scanf("%d", &num[i]);
   }

   for (int i = 0; i < 5; i++) {
      for (int j = i + 1; j < 5; j++) {
         if (num[i] > num[j]) {
                   temp = num[i];
            num[i] = num[j];
            num[j] = temp;
         }
      }
   }

   printf("\nNumbers in ascending order:\n");
   for (int i = 0; i < 5; i++) {
      printf("%d\n", num[i]);
   }

   return 0;
}
```

Que.16 Accept 5 numbers from user and perform sum of array

```
#include <stdio.h>

int main() {
   int num[5],sum=0;

   printf("Enter 5 numbers:\n");
   for (int i = 0; i < 5; i++) {
      printf("Number %d: ", i + 1);
      scanf("%d", &num[i]);
   }

   for (int i = 0; i < 5; i++) {
      sum += num[i];
   }

   printf("\nThe sum of the numbers is: %d\n", sum);

   return 0; }
```

Que.17 WAP to show difference between Structure and Union.

```
#include <stdio.h>

struct MyStruct {
   int intVal;
   float floatVal;
   char charVal;
};


union MyUnion {
   int intVal;
   float floatVal;
   char charVal;
};

int main() {

   struct MyStruct s;
   s.intVal = 10;
   s.floatVal = 20.5;
   s.charVal = 'A';

   union MyUnion u;
   u.intVal = 10;
   u.floatVal = 20.5;
   u.charVal = 'A';


   printf("Structure values:\n");
   printf("intValue: %d\n", s.intVal);
   printf("floatValue: %.2f\n", s.floatVal);
   printf("charValue: %c\n", s.charVal);


   printf("\nUnion values :\n");
   printf("intValue: %d\n", u.intVal);
   printf("floatValue: %.2f\n", u.floatVal);
   printf("charValue: %c\n", u.charVal);


   printf("\nSize of Structure: %zu bytes\n", sizeof(s));
   printf("Size of Union: %zu bytes\n", sizeof(u));

   return 0; }
```