FCP LAB MANUAL

GROUP A:-

Q 1) Program to read three numbers and find the biggest of three.

```
#include <stdio.h>

int main() {
    int num1, num2, num3;

    // Input the three numbers
    printf("Enter three numbers: ");
    scanf("%d %d %d", &num1, &num2, &num3);

    // Compare the numbers to find the largest
    if (num1 >= num2 && num1 >= num3) {
         printf("The biggest number is: %d\n", num1);
     } else if (num2 >= num1 && num2 >= num3) {
         printf("The biggest number is: %d\n", num2);
    } else {
        printf("The biggest number is: %d\n", num3);
    }

    return 0;
}
```

Q2) Program to find the area of a triangle using three sides of triangle.

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

int main() {
    float side1, side2, side3, area, s;

    // Input the three sides of the triangle
    printf("Enter three sides of the triangle: ");
    scanf("%f %f %f", &side1, &side2, &side3);

    // Calculate the semi-perimeter (s)
    s = (side1 + side2 + side3) / 2;

    // Calculate the area using Heron's formula
    area = sqrt(s * (s - side1) * (s - side2) * (s - side3));

    // Display the area
    printf("The area of the triangle is: %.2f\n", area);

    return 0;
}
```

Q3) Program to check for prime number.

```
#include <stdio.h>

int main() {
    int num, i, isPrime = 1;
    printf("Enter a number: ");
    scanf("%d", &num);

    // Handle special cases for numbers less than 2
    if (num <= 1) {
        isPrime = 0; // Numbers less than or equal to 1 are not prime
    } else {
        // Check if the number is divisible by any number from 2 to num-1
        for (i = 2; i <= num / 2; i++) {
            if (num % i == 0) {
                isPrime = 0; // If divisible, it's not a prime number
                break;
            }
        }
    }
    if (isPrime) {
        printf("%d is a prime number.\n", num);
    } else {
        printf("%d is not a prime number.\n", num);
    }

    return 0;
}
```

Q4) Program to generate n Fibonacci numbers

```
#include <stdio.h>

int main() {
    int n, i;
    int t1 = 0, t2 = 1, nextTerm;

    // Input the number of terms in the Fibonacci sequence
    printf("Enter the number of Fibonacci terms: ");
    scanf("%d", &n);

    // Display the first two Fibonacci terms
    printf("Fibonacci Series: ");

    // Special case for n = 1 (only the first term)
    if (n >= 1) {
        printf("%d ", t1);
    }
    if (n >= 2) {
        printf("%d ", t2);
    }

    // Generate and display the Fibonacci sequence
    for (i = 3; i <= n; i++) {
        nextTerm = t1 + t2;
        printf("%d ", nextTerm);
        t1 = t2;
        t2 = nextTerm;
    }

    printf("\n");
    return 0;
}
```

Q5) Program to read a multi - digit number find the sum of the digits, reverse the number and check it for palindrome.

```
#include <stdio.h>

int main() {
    int num, originalNum, sum = 0, reverse = 0, digit;
    printf("Enter a multi-digit number: ");
    scanf("%d", &num);

    originalNum = num;  // Save the original number for palindrome check

    // Calculate the sum of digits
    while (num != 0) {
        digit = num % 10;  // Get the last digit
        sum += digit;      // Add the digit to sum
        num /= 10;         // Remove the last digit
    }

    // Reset the number to original for reversing
    num = originalNum;

    // Reverse the number
    while (num != 0) {
        digit = num % 10;  // Get the last digit
        reverse = reverse * 10 + digit;  // Build the reversed number
        num /= 10;         // Remove the last digit
    }

    // Check if the number is a palindrome
    if (originalNum == reverse) {
        printf("%d is a palindrome number.\n", originalNum);
    } else {
        printf("%d is not a palindrome number.\n", originalNum);
    }
    printf("The sum of the digits is: %d\n", sum);

    return 0;
}
```

Q6) Program to read numbers from keyboard continuously till the user presses 999 and to find the sum of only positive numbers.

```
#include <stdio.h>

int main() {
    int num, sum = 0;

    // Continuously read numbers until the user enters 999
    while (1) {
        printf("Enter a number (999 to stop): ");
        scanf("%d", &num);

        // If the number is 999, break the loop
        if (num == 999) {
            break;
        }

        // Add only positive numbers to the sum
        if (num > 0) {
            sum += num;
        }
    }
    printf("The sum of positive numbers is: %d\n", sum);

    return 0;
}
```

Q7) Program to accept student name and marks in three subjects. Find the total marks, average and grade (depending on the average marks).

```
#include<stdio.h>
int main()
{
 int s1,s2,s3,total;
 float average;
 printf("Enter three subject marks:\n");
scanf("%d%d%d",&s1,&s2,&s3);
total=s1+s2+s3;
 average=total/3;
 printf("Total= %d\nAverage= %f",total,average);

return 0;
}
```

Q8) Program to find the roots of quadratic equation (Demonstration of switch Statement)

```
#include <stdio.h>
#include <math.h>  // For sqrt() function

// Function to find the roots of the quadratic equation
void findRoots(float a, float b, float c) {
    float D, root1, root2, realPart, imagPart;

    // Calculate the discriminant
    D = b * b - 4 * a * c;

    // Use switch statement to decide the nature of roots
    switch (D > 0) {
        case 1:  // If discriminant is positive (D > 0)
            root1 = (-b + sqrt(D)) / (2 * a);
            root2 = (-b - sqrt(D)) / (2 * a);
            printf("Roots are real and distinct.\n");
            printf("Root 1 = %.2f\n", root1);
            printf("Root 2 = %.2f\n", root2);
            break;

        case 0:  // If discriminant is zero (D = 0)
            root1 = root2 = -b / (2 * a);
            printf("Roots are real and equal.\n");
            printf("Root 1 = Root 2 = %.2f\n", root1);
            break;

        default:  // If discriminant is negative (D < 0)
            realPart = -b / (2 * a);
            imagPart = sqrt(-D) / (2 * a);
            printf("Roots are complex and different.\n");
            printf("Root 1 = %.2f + %.2fi\n", realPart, imagPart);
            printf("Root 2 = %.2f - %.2fi\n", realPart, imagPart);
            break;
    }
}

int main() {
    float a, b, c;

    // Input coefficients
    printf("Enter the coefficients a, b, and c of the quadratic equation (ax^2 + bx + c
= 0):\n");
    printf("a: ");
    scanf("%f", &a);
    printf("b: ");
    scanf("%f", &b);
    printf("c: ");
    scanf("%f", &c);

    // Ensure that 'a' is not zero (as it would not be a quadratic equation)
    if (a == 0) {
        printf("The value of 'a' should not be zero.\n");
    } else {
        // Call the function to find and display the roots
        findRoots(a, b, c);
    }

    return 0;
}
```

Q9) Program to multiply two matrices.

```
#include<stdio.h>
#include<stdlib.h>
int main(){
int a[10][10],b[10][10],mul[10][10],r,c,i,j,k;
system("cls");
printf("enter the number of row=");
scanf("%d",&r);
printf("enter the number of column=");
scanf("%d",&c);
printf("enter the first matrix element=\n");
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
scanf("%d",&a[i][j]);
}
}
printf("enter the second matrix element=\n");
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
scanf("%d",&b[i][j]);
}
}

printf("multiply of the matrix=\n");
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
mul[i][j]=0;
for(k=0;k<c;k++)
{
mul[i][j]+=a[i][k]*b[k][j];
}
}
}
//for printing result
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
printf("%d\t",mul[i][j]);
}
printf("\n");
}
return 0;
}
```

Q10) Program to find largest and smallest element in a list of ‘n’ elements (Demonstration of one- dimensional array).

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

```c
int main() {
    int n, i, largest, smallest;


    printf("Enter the number of elements: ");
    scanf("%d", &n);

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


    largest = smallest = arr[0];


    for (i = 1; i < n; i++) {
        if (arr[i] > largest) {
            largest = arr[i];
        }
        if (arr[i] < smallest) {
            smallest = arr[i];
        }
    }


    printf("\nLargest element: %d\n", largest);
    printf("Smallest element: %d\n", smallest);

    return 0;
}
```

GROUP B:-

Q1) Program to accept ‘n’ and find the sum of the series 1! + 3! + 5! …… + n!

```
#include <stdio.h>

int main() {
    int n;
    long long int sum = 0;

    // Input the value of n
    printf("Enter the value of n: ");
    scanf("%d", &n);

    // Loop through odd numbers from 1 to n
    for (int i = 1; i <= n; i += 2) {
        long long int fact = 1;

        // Calculate factorial of the current odd number
        for (int j = 1; j <= i; j++) {
            fact *= j;
        }

        // Add the factorial to the sum
        sum += fact;
    }

    // Output the sum of the series
    printf("Sum of the series is: %lld\n", sum);

    return 0;
}
```

Q2) Program to find whether a given string is palindrome or not (reverse a string using pointers)

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

```
int main() {
    char str[100], *start, *end, temp;
    int i, isPalindrome = 1;

    // Input the string
    printf("Enter a string: ");
    scanf("%s", str);  // This reads the string without spaces.

    // Initialize pointers to the start and end of the string
    start = str;
    end = str;

    // Move the 'end' pointer to the last character of the string
    while (*end != '\0') {
        end++;
    }
    end--;  // Move back to the last character

    // Compare characters from start and end using pointers
    while (start < end) {
        if (*start != *end) {
            isPalindrome = 0;  // If characters don't match, it's not a palindrome
            break;
        }
        start++;  // Move the start pointer forward
        end--;    // Move the end pointer backward
    }

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

    return 0;
}
```

Q3) Program to transpose a matrix of order N x M and check whether it is symmetric or not.

```
#include <stdio.h>

int main() {
    int N, M, i, j;

    // Input the dimensions of the matrix
    printf("Enter the number of rows (N): ");
    scanf("%d", &N);
    printf("Enter the number of columns (M): ");
    scanf("%d", &M);

    int matrix[N][M], transpose[M][N];

    // Input the elements of the matrix
    printf("Enter the elements of the matrix:\n");
    for (i = 0; i < N; i++) {
        for (j = 0; j < M; j++) {
            scanf("%d", &matrix[i][j]);
        }
    }

    // Transpose the matrix
    for (i = 0; i < N; i++) {
        for (j = 0; j < M; j++) {
            transpose[j][i] = matrix[i][j];
        }
    }

    // Display the transposed matrix
    printf("\nThe transposed matrix is:\n");
    for (i = 0; i < M; i++) {
        for (j = 0; j < N; j++) {
            printf("%d ", transpose[i][j]);
        }
        printf("\n");
    }
    // Check whether the matrix is symmetric or not
    if (N == M) {  // A matrix can only be symmetric if it's square (N == M)
        int isSymmetric = 1;
        for (i = 0; i < N; i++) {
            for (j = 0; j < N; j++) {
                if (matrix[i][j] != transpose[i][j]) {
                    isSymmetric = 0;  // If any element doesn't match, it's not symmetric
                    break;
                }
            }
            if (isSymmetric == 0) {
                break;
            }
        }

        if (isSymmetric == 1) {
            printf("\nThe matrix is symmetric.\n");
        } else {
            printf("\nThe matrix is not symmetric.\n");
        }
    } else {
        printf("\nThe matrix is not square, so it cannot be symmetric.\n");
    }

    return 0;
}
```

Q4) Program to add two matrices using pointers.

```
#include <stdio.h>

int main() {
    int i, j, row, col;

    // Input the dimensions of the matrices
    printf("Enter the number of rows: ");
    scanf("%d", &row);
    printf("Enter the number of columns: ");
    scanf("%d", &col);
    int matrix1[row][col], matrix2[row][col], sum[row][col];

    // Input elements for the first matrix
    printf("Enter elements of the first matrix:\n");
    for (i = 0; i < row; i++) {
        for (j = 0; j < col; j++) {
            scanf("%d", &matrix1[i][j]);
        }
    }

    // Input elements for the second matrix
    printf("Enter elements of the second matrix:\n");
    for (i = 0; i < row; i++) {
        for (j = 0; j < col; j++) {
            scanf("%d", &matrix2[i][j]);
        }
    }

    // Add the two matrices using pointers
    for (i = 0; i < row; i++) {
        for (j = 0; j < col; j++) {
            *(sum[i] + j) = *(matrix1[i] + j) + *(matrix2[i] + j);  // Pointer arithmetic
        }
    }

    // Display the sum matrix
    printf("Sum of the two matrices is:\n");
    for (i = 0; i < row; i++) {
        for (j = 0; j < col; j++) {
            printf("%d ", sum[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

Q5) Program to read a string and to find the number of alphabets, digits, vowels, consonants, spaces and special characters.

```
#include <stdio.h>
#include <ctype.h>  // For checking characters

int main() {
    char str[100];
    int alphabets = 0, digits = 0, vowels = 0, consonants = 0, spaces = 0,
specialChars = 0;
    int i = 0;

    // Input the string
    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);  // Using fgets to read string with spaces

    // Loop through each character in the string
    while (str[i] != '\0') {
        if (isalpha(str[i])) {  // Check if the character is an alphabet
            alphabets++;

            // Check if the alphabet is a vowel
            if (str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u' ||
                str[i] == 'A' || str[i] == 'E' || str[i] == 'I' || str[i] == 'O' || str[i] == 'U') {
                vowels++;
            } else {  // If it's not a vowel, it's a consonant
                consonants++;
            }
        } else if (isdigit(str[i])) {  // Check if the character is a digit
            digits++;
        } else if (isspace(str[i])) {  // Check if the character is a space
            spaces++;
        } else {  // If it's not an alphabet, digit, or space, it's a special character
            specialChars++;
        }
        i++;
    }

    // Output the results
    printf("Alphabets: %d\n", alphabets);
    printf("Digits: %d\n", digits);
    printf("Vowels: %d\n", vowels);
    printf("Consonants: %d\n", consonants);
    printf("Spaces: %d\n", spaces);
    printf("Special characters: %d\n", specialChars);

    return 0;
}
```

Q6) Program to display the first ‘n’ Fibonacci numbers to generate the nth Fibonacci number.

```
#include <stdio.h>
int main() {
    int n, first = 0, second = 1, next;
    printf("Enter the value of n: ");
    scanf("%d", &n);
    printf("Fibonacci series up to %d terms: \n", n);
    for (int i = 1; i <= n; i++) {
        if (i == 1) {
            printf("%d ", first);
            continue;
        }
        if (i == 2) {
            printf("%d ", second);
            continue;
        }
        next = first + second;
        printf("%d ", next);
        first = second;
        second = next;
    }
    printf("\n");
    return 0;
}
```

Q7) Program to find the GCD of ‘n’ integers to compute the GCD of two integers.

```
#include <stdio.h>
int gcd(int a, int b) {
    while (b != 0) {
        int temp = b;
        b = a % b;
        a = temp;
    }
    return a;
}
int main() {
    int num1, num2;
    printf("Enter two integers: ");
    scanf("%d %d", &num1, &num2);
    printf("The GCD of %d and %d is: %d\n", num1, num2, gcd(num1, num2));
    return 0;
}
```

Q8) Program to enter the information of n students (name, register number, marks in three subjects). Compute and print the result of all students. For passing, student should get at least 35 in each subject, otherwise result is “FAIL”. If the student passes and if percentage >= 70, result is DISTINCTION; If percentage is < 70 and >= 60, result is FIRST CLASS; if percentage is < 60 and >=50, result is SECOND CLASS; otherwise, result is PASS CLASS. Get the output of all students in a tabular form with proper column headings.

```
#include <stdio.h>

int main() {
    int n;
    printf("Enter the number of students: ");
    scanf("%d", &n);
    char name[100];
    int regNo, marks1, marks2, marks3;
    float percentage;
    for (int i = 0; i < n; i++) {
        printf("\nEnter details for student %d\n", i + 1);
        printf("Enter student's name: ");
        getchar();
        fgets(name, sizeof(name), stdin);

        printf("Enter register number: ");
        scanf("%d", &regNo);

        printf("Enter marks in three subjects: ");
        scanf("%d %d %d", &marks1, &marks2, &marks3);
        if (marks1 >= 35 && marks2 >= 35 && marks3 >= 35) {
            // Calculate percentage
            percentage = (marks1 + marks2 + marks3) / 3.0;
            if (percentage >= 70) {
                printf("Result: DISTINCTION\n");
            } else if (percentage >= 60) {
                printf("Result: FIRST CLASS\n");
            } else if (percentage >= 50) {
                printf("Result: SECOND CLASS\n");
            } else {
                printf("Result: PASS CLASS\n");
            }
        } else {
            printf("Result: FAIL\n");
        }
        printf("Name: %s", name);
        printf("Register No.: %d\n", regNo);
        printf("Marks: %d %d %d\n", marks1, marks2, marks3);
        printf("Percentage: %.2f%%\n", percentage);
    }

    return 0;
}
```

Q9) Program to prepare the pay slip of ‘n’ employees. Input the employee’s name, employee number and basic pay. Calculate the DA, HRA, PF, PT, Gross Pay and Net Pay as follows:
If Basic < 40000, DA = 50% of Basic, HRA = 12% of Basic, PF = 12% of Gross Pay, PT = 250.Otherwise DA = 40% of Basic, HRA = 10% of Basic, PF = 13% of Gross, PT= 300. Gross Pay = Basic + DA + HRA and Net Pay = Gross Pay – PF – PT.

```
#include <stdio.h>

int main() {
    int n;
    printf("Enter the number of employees: ");
    scanf("%d", &n);


    char name[100];
    int empNumber;
    float basic, DA, HRA, PF, PT, grossPay, netPay;


    for (int i = 0; i < n; i++) {
        printf("\nEnter details for employee %d\n", i+1);


        printf("Enter employee's name: ");
        getchar();
        fgets(name, sizeof(name), stdin);

        printf("Enter employee number: ");
        scanf("%d", &empNumber);

        printf("Enter basic pay: ");
        scanf("%f", &basic);


        if (basic < 40000) {
            DA = 0.50 * basic;
            HRA = 0.12 * basic;
            PT = 250;
        } else {
            DA = 0.40 * basic;              HRA = 0.10 * basic;
            PT = 300;
        }


        grossPay = basic + DA + HRA;
        PF = 0.12 * grossPay; // PF = 12% of Gross Pay


        netPay = grossPay - PF - PT;


        printf("\nPay Slip for Employee: %s", name);
        printf("Employee Number: %d\n", empNumber);
        printf("Basic Pay: %.2f\n", basic);
        printf("DA: %.2f\n", DA);
        printf("HRA: %.2f\n", HRA);
        printf("PF: %.2f\n", PF);
        printf("PT: %.2f\n", PT);
        printf("Gross Pay: %.2f\n", grossPay);
        printf("Net Pay: %.2f\n", netPay);
    }

    return 0;
}
```

Q10) Write user-defined functions to (a) find the length of a string (b) concatenate two strings. Call these functions in the main program.

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

int main() {
    char dest[100] = "C";
    char src[100] = "Programming";


    strcat(dest, src);

    printf("Concatenated String: %s\n", dest);
    return 0;
}
```