

RAMAIAH INSTITUTE OF TECHNOLOGY
DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING
INTRODUCTION TO C PROGRAMMING - ESC135

INTEGRATEDLAB

Session -1

Creating and Running Simple C Programs:

1. Write a program to swap two numbers without using a temporary variable.

Sol:

```
#include<stdio.h>
int main()
{
    int a=10, b=20;
    printf("Before swap a=%d b=%d",a,b);
    a=a+b;//a=30 (10+20)
    b=a-b;//b=10 (30-20)
    a=a-b;//a=20 (30-10)
    printf("\nAfter swap a=%d b=%d",a,b);
    return 0;
}
```

2. Write a program to convert degrees Fahrenheit into degrees Celsius.

```
#include <stdio.h>

intmain()
{
    float celsius, fahrenheit;
    /* Input temperature in fahrenheit */
    printf("Enter temperature in Fahrenheit: ");
    scanf("%f", &fahrenheit);
    /* Fahrenheit to celsius conversion formula */
    celsius = (fahrenheit - 32) * 5 / 9;
    /* Print the value of celsius */
    printf("%.2f Fahrenheit = %.2f Celsius", fahrenheit, celsius);
    return 0;
}
```

3. Write a program to calculate the area and circumference of a circle, using PI as symbolic constant.

```
#include<stdio.h>
// #define PI 3.14 can be used as global declaration

intmain()
{
    int rad;
    float PI = 3.14, area, ci;
```

```

        printf("\nEnter radius of circle: ");
        scanf("%d", &rad);
        area = PI * rad * rad;
        printf("\nArea of circle : %f ", area);
        ci = 2 * PI * rad;
        printf("\nCircumference : %f ", ci);
        return (0);
    }

```

4. Write a C program to calculate the bill amount for an item given its quantity sold, value, discount (declare as constant) and tax (declare as constant).

```

#include <stdio.h>
#define discount 0.02
#define tax 0.01
int main()
{
    float quantity, price, amount, tot_amount;
    printf("Enter quantity and price:");
    scanf("%f%f", &quantity, &price);
    amount = quantity * price;
    tot_amount = amount + (amount * tax) - (amount * discount);
    printf("Total amount to be paid is %f", tot_amount);
    return 0;
}

```

5. Write a C program to read two floating point numbers. Add these numbers and assign the result to an integer. Finally display the value of all three variables.

```

#include <stdio.h>
int main()
{
    float f1, f2;
    int total;
    printf("Enter your floating point number1:");
    scanf("%f", &f1);
    printf("Enter your floating point number2:");
    scanf("%f", &f2);
    total = f1 + f2;
    printf("The sum of %f and %f is %d", f1, f2, total);
    return 0;
}

```

6. Write a program to calculate the salary of an employee, given his basic pay (to be entered by user), HRA = 10% of the basic pay, TA = 5% of basic pay. Define HRA and TA as symbolic constants and calculate the salary of the employee. [Salary = Basic Pay + HRA + TA]

```
#include<stdio.h>
int main()
{
    float TA,HRA, Salary;
    int BP;
    printf("Enter your Basic Pay:");
    scanf("%d",&BP);
    TA= 0.05*BP;
    HRA = 0.1*BP;
    Salary = BP+TA+HRA;
    printf("Total Salary is: %f ",Salary);
    return 0;
}
```

Session -2

Creating and Running C Programs on Making Decisions - 1:

1. Write a program to find the given year is a leap year or not.

```
#include <stdio.h>
intmain()
{
    int year;
    printf("Enter a year \n");
    scanf("%d",&year);
    if (((year % 4 == 0) && (year % 100!= 0)) || (year%400 == 0))
        printf("%d is a leap year", year);
    else
        printf("%d is not a leap year", year);
    return 0;
}
```

2. Write a program to determine whether the given character is a vowel or not.

```
#include <stdio.h>
intmain()
{
    char c;
    intlowercase_vowel, uppercase_vowel;
    printf("Enter an alphabet: ");
    scanf("%c", &c);
    // evaluates to 1 if variable c is a lowercase vowel
    lowercase_vowel = (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u');
    // evaluates to 1 if variable c is a uppercase vowel
    uppercase_vowel = (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U');
```

```

        // evaluates to 1 (true) if c is a vowel
        if (lowercase_vowel || uppercase_vowel)
            printf("%c is a vowel.", c);
        else
            printf("%c is a consonant.", c);
    return 0;
}

```

3. Write a program to identify whether the given character is an alphabet, digit, whitespace or punctuation using 'simple if'

```

#include <stdio.h>
#include <ctype.h>
int main()
{
    char ch;
    /* Input character from user */
    printf("Enter any character: ");
    scanf("%c", &ch);
    /* Alphabet check */
    if (isalpha(ch))
    {
        printf("%c is alphabet.", ch);
    }
    else if (isdigit(ch))
    {
        printf("%c is digit.", ch);
    }
    else if (ispunct(ch))
    {
        printf("%c is punctuation.", ch);
    }
    elseif (isspace(ch))
    {
        printf("This is a whitespace");
    }
    else
    {
        printf("Invalid Character");
    }
    return 0;
}

```

4. Write a C Program to find whether the given triangle is equilateral, isosceles or scalene.

```

#include <stdio.h>
int main()
{
    int side1, side2, side3;

```

```

printf("Enter sides of triangle:");
scanf("%d%d%d",&side1,&side2,&side3);
if(side1 == side2 && side2 == side3)
    printf("The Given Triangle is equilateral");
else if(side1 == side2 || side2 == side3 || side3 == side1)
    printf("The given Triangle is isosceles");
else
    printf("The given Triangle is scalene");
return 0;
}

```

5. Write a program to find the greatest of two numbers using conditional operator.

```

#include<stdio.h>
intmain()
{
    inta,b,max;
    printf("Enter values for a and b : ");
    scanf("%d%d", &a, &b);
    max = a>b ? a : b;
    printf("Larger of %d and %d is %d \n", a, b, max);
    return 0;
}

```

6. Write a program, to find the greatest of three numbers using nested –if

```

#include<stdio.h>

void main()
{
    int a, b, c;
    printf("Enter three numbers\n");
    scanf("%d %d %d", &a, &b, &c);
    if(a > b)
    {
        if(a > c)
            printf("a: %d is largest\n", a);
        else
            printf("b: %d is largest\n", b);
    }
    else if(b > c)
        printf("b: %d is largest\n", b);
    else
        printf("c: %d is largest\n", c);
}

```

Session-3:

Creating and Running C Programs on Making Decision – 2

1. Find the roots of a quadratic equation using Switch statement. ($D=0, D>0, D<0$)

```
/*C program to find all roots of a quadratic equation using switch
case */

#include <stdio.h>
#include <math.h> /* Used for sqrt() */
int main()
{
    float a, b, c;
    float root1, root2, imaginary, real, imag;
    float discriminant;

    printf("Enter values of a, b, c of quadratic equation ");
    scanf("%f%f%f", &a, &b, &c);

    /* Calculate discriminant */
    discriminant = (b * b) - (4 * a * c);

    /* Compute roots of quadratic equation based on the nature of
    discriminant */

    switch(discriminant > 0)
    {
        case 1:
            root1 = (-b + sqrt(discriminant)) / (2 * a);
            root2 = (-b - sqrt(discriminant)) / (2 * a);
            printf("Two distinct and real roots exists: %.2f and %.2f", root1,
            root2);
            break;

        case 0:
            switch(discriminant < 0)
            {
                case 1:
                    printf("Two distinct complex roots exists \n");
                    real = -b/(2*a);
                    imag = sqrt(fabs(discriminant))/(2*a);
                    printf("Root1 = %.2f + i%.2f", real, imag \n");
                    printf("Root1 = %.2f - i%.2f", real, imag \n");
                    break;
                case 0:
                    root1 = root2 = -b / (2 * a);
                    printf("Two equal and real roots exists: %.2f and %.2f",
                    root1, root2);
                    break;
            }
    }
    return 0;
}
```

2. Write a program that accepts a number from 1 to 10. Print whether the number is even or odd using switch case construct.

```
/*C program to check Even or Odd number using switch case */  
  
#include <stdio.h>  
  
intmain()  
{intnum;  
  
    /* Input a number from user */  
  
    printf("Enter any number to check even or odd: ");  
  
    scanf("%d",&num);  
  
    switch(num % 2)  
    {case 0:  
        printf("Number is Even");  
        break;  
        case 1:  
        printf("Number is Odd");  
        break;  
    }return 0;  
}
```

3. Write a C program to enter the marks of a student in four subjects. Then calculate the total, aggregate and display the grades obtained by the student (Using SWITCH).

```
#include <stdio.h>  
  
intmain()  
{  
    intphy, chem, math, comp, sum, agg;  
    float per;  
  
    /* Input marks of four subjects from user */  
    printf("Enter four subjects marks: ");  
    scanf("%d%d%d%d", &phy, &chem, &math, &comp);  
    /* Calculate percentage */  
    sum = phy + chem + math + comp;  
    per =sum / 4.0;
```

```

    //agg = sum / 4;
    agg = per; // auto conversion from float to int
    printf("Percentage = %.2f\n", per);
    printf("aggregate = %d\n", agg);

    switch(agg/10)
    {
        case 10:
        case 9: printf("Grade A \n");
                break;
        case 8: printf("Grade B \n");
                break;
        case 7: printf("Grade C \n");
                break;
        case 6: printf("Grade D \n");
                break;
        case 5: printf("Grade Pass \n");
                break;
        default: printf("Fail \n");
    }
    return 0;
}

```

4. Write a program to calculate the parking charges of a vehicle. Enter the type of vehicle as a character (like c for car, b for bus etc) and the number of hours. Then calculate the charges as given below. [Program must be implemented using Switch and else-if]
- Truck / Bus – Rs 20 per hour
 - Car – Rs 10 per hour
 - Scooter/ Cycle/ Motor cycle – Rs 5 per hour

```

#include<stdio.h>
void main()
{
    int car=10,bus=20,Sc=5,noofhours, charges=0;
    char choice;
    printf("Please enter vehicle details c-car b-bus and s-scooter \n");
    scanf("%c", &choice);
    printf("Please enter no of hours");
    scanf("%d",&noofhours);

    switch(choice)
    {
        case 'c': charges=car*noofhours;
        printf("Parking charges for your car = %d \n",charges);
                break;
    }
}

```



```
    case 'b': charges=bus*noofhours;
printf("Parking charges for your bus = %d \n",charges);
        break;
    case 's': charges=Sc*noofhours;
printf("Parking charges for your Scooter = %d \n",charges);
        break;
    default: printf("Invalid Choice \n");
}
}
```

Session-4:

Creating and Running C Programs on Repetition or Loops:

1. Write a program to calculate the sum of numbers from m to n using FOR loop.

```
#include <stdio.h>

intmain()
{
    inti, m,n, sum=0;

    printf("Enter lower limit: ");
    scanf("%d", &m);

    printf("Enter upper limit: ");
    scanf("%d", &n);

    /* Find sum of all numbers */
    for(i=m; i<=n; i++)
    {
        sum += i;
    }
    printf("Sum of numbers between %d and %d = %d", m, n, sum);
    return 0;
}
```

2. Write a program to generate the Fibonacci series up to 'nth' given number using WHILE loop.

```
#include<stdio.h>

intmain()
{
    int f1=0,f2=1,nextTerm,i=3,n;
    printf("Enter a Positive Number \n:");
    scanf("%d",&n);

    if (n == 1)
        printf("%d", f1);
    else
        if (n>=2)
        {
            printf("%d\t%d",f1,f2);
            nextTerm = f1 + f2;

            for(i=0; i<n-2;i++)
            {
```

```

printf("\t%d", nextTerm);
    f1= f2;
    f2= nextTerm;
nextTerm = f1+ f2;
}
return 0;
}

```

3. Write a program to generate prime numbers within a given range. [using for loop].

```

#include <stdio.h>
#include <stdlib.h>

void main()
{
int num1, num2, i, j, flag, temp;

printf("Enter the value of num1 and num2 \n");
scanf("%d %d", &num1, &num2);

printf("Prime numbers are \n");

    temp = num1;
    if (num1 % 2 == 0)
    {
        num1++;
    }

    for (i = num1; i<= num2; i = i + 2)
    {
        flag = 0;
        for (j = 2; j <= i / 2; j++)
        {
            if ((i % j) == 0)
            {
                flag = 1;
                break;
            }
        }
        if (flag == 0)
        {
            printf("%d\n", i);
        }
    }
}

```

4. Write a program to check whether the given number is a palindrome or not using DO - WHILE loop

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,a,r,s=0;
    printf("\n Enter The Number:");
    scanf("%d",&n);
    a=n;
    //LOOP TO FIND REVERSE OF A NUMBER

    do
    {
        r=n%10;
        s=s*10+r;
        n=n/10;
    }while(n>0);

    /* CHECKING IF THE NUMBER ENTERED AND THE REVERSE NUMBER IS EQUAL
    OR NOT */
    if(a==s)
    {
        printf("\n %d is a Palindrome Number",a);
    }
    else
    {
        printf("\n %d is a not Palindrome Number",a);
    }
}
```

Session -5
Creating and Running Simple on One Dimensional Array:

1. Write a C program to find the average of all elements of a 1D array.

```
#include<stdio.h>

int main()
{
    int array[10]={1,2,3,4,5,6,7,8,9,0};
    int sum, a, loop ;
    float avg;

    sum = avg = 0;

    for(loop =0; loop <10; loop++)
    {
        sum = sum + array[loop];
    }

    avg=(float)sum / loop;
    printf("Average of array values is %.2f",avg);

    return0;
}
```

2. Write a C program to search for an element using Linear Search

```
#include <stdio.h>

int main()
{
    int array[100], key, c, number;
    printf("Enter the number of elements in array\n");
    scanf("%d",&number);
    printf("Enter %d numbers\n", number);
    for ( c = 0 ; c < number ; c++ )
        scanf("%d",&array[c]);
    printf("Enter the number to search\n");
    scanf("%d",&key);
```

```

for ( c = 0 ; c < number ; c++ )
{
    if ( array[c] == key ) /* if required element found */
    {
        printf("%d is present at location %d.\n", key, c+1);
        break;
    }
}

if ( c == number )
printf("%d is not present in array.\n", search);
return 0;
}

```

3. The books are placed in a random order in a library. Write a C program to sort the books based on ISBN (Bubble Sort).

```

#include <stdio.h>

int main()
{
    int array[100], n, c, d, swap;
    printf("Enter number of elements\n");
    scanf("%d", &n);
    printf("Enter %d integers\n", n);
    for (c = 0; c < n; c++)
        scanf("%d", &array[c]);
    for (c = 0 ; c < n - 1; c++)
    {
        for (d = 0 ; d < n - c - 1; d++)
        {
            if (array[d] > array[d+1]) /* For decreasing order use '<' */
            {
                swap = array[d];
                array[d] = array[d+1];
                array[d+1] = swap;
            }
        }
    }
}

```

4. Write a C program to search for a book based on the ISBN whether the book is present or not (Binary Search).

```
#include<stdio.h>

void main()
{
    int a[30],i,num,key,low,mid,high;
    printf("\n enter the no of elements");
    scanf("%d",&n);
    printf("\n enter the elements : ");
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }

    printf("\n enter the key element to be searched\n");
    scanf("%d",&key);
    low=0;
    high=n-1;
    while(low <= high)
    {
        mid=(low+high)/2;
        if(a[mid]==key)
        {
            printf("element %d is found at %d position :",key,mid+1);
            exit(0);
        }
        else if (a[mid] > key)
            high = mid-1;
        else low = mid+1;
    }
    printf("element not found\n");
}
```

Session -6
Creating and Running Simple on Two Dimensional Array:

1. Write a C program to find the sum of two matrices using functions

```
#include <stdio.h>

int rows, columns;

/* adds two matrices and stores the output in third matrix */
void matrixAddition(int mat1[][10], int mat2[][10], int mat3[][10]) {
    int i, j;
    for (i = 0; i < rows; i++) {
        for (j = 0; j < columns; j++) {
            mat3[i][j] = mat1[i][j] + mat2[i][j];
        }
    }
    return;
}

int main() {
    int matrix1[10][10], matrix2[10][10];
    int matrix3[10][10], i, j;

    /* get the number of rows and columns from user */
    printf("Enter the no of rows and columns(<=10):");
    scanf("%d%d", &rows, &columns);

    if (rows > 10 || columns > 10) {
        printf("No of rows/columns is greater than 10\n");
        return 0;
    }

    /* input first matrix */
    printf("Enter the input for first matrix:");
    for (i = 0; i < rows; i++) {
        for (j = 0; j < columns; j++) {
            scanf("%d", &matrix1[i][j]);
        }
    }

    /* input second matrix */
    printf("Enter the input for second matrix:");
    for (i = 0; i < rows; i++) {
        for (j = 0; j < columns; j++) {
            scanf("%d", &matrix2[i][j]);
        }
    }
}
```



```

/* matrix addition */
matrixAddition(matrix1, matrix2, matrix3);

/* print the results */
printf("\nResult of Matrix Addition:\n");
for (i = 0; i < rows; i++) {
    for (j = 0; j < columns; j++) {
        printf("%5d", matrix3[i][j]);
    }
    printf("\n");
}
return 0;
}

```

2. Write a C program to find the product of two matrices

```

#include<stdio.h>
#include<stdlib.h>
void main()
{
    int i,j,m,n,p,q,k,a[5][5],b[5][5],c[5][5];
    printf("Enter the order of Matrix A\n");
    scanf("%d%d",&m,&n);          /* Read the size of first matrix */
    printf("Enter the order of Matrix B\n");
    scanf("%d%d",&p,&q);          /* Read the size of second matrix */
    if(n!=p)                      /* Check whether multiplication is possible or not */
    {
        printf("Multiplication is not possible\n");
        exit(0);
    }
    printf("Enter the elements of matrix A\n");
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            scanf("%d",&a[i][j]);    /* Read the elements of matrix A in row major order*/
        }
    }
}

```

```

printf("Enter the elements of matrix B\n");
for(i=0;i<p;i++)
{ for(j=0;j<q;j++)
    {
        scanf("%d",&b[i][j]);
    }
}
/* Multiply the matrices */
For (i=0;i<m;i++)
{
    for(j=0;j<q;j++)
    {
        c[i][j]=0;
        for(k=0;k<n;k++)
            c[i][j]+=a[i][k]*b[k][j];
    }
}

printf("Product matrix is\n");
for(i=0;i<m;i++)
{
    for(j=0;j<q;j++)
    {
        printf("%d\t",c[i][j]);          /* Display product matrix */
    }
    printf("\n");
}
}/*end of main*/

```

3. Write a C program to find the transpose of a given matrix.

```

#include <stdio.h>
int main()
{
    int a[10][10], transpose[10][10], r, c;
    printf("Enter rows and columns: ");

```

```

scanf("%d %d", &r, &c);
// assigning elements to the matrix
printf("\nEnter matrix elements:\n");
for (int i = 0; i < r; ++i)
    for (int j = 0; j < c; ++j) {
        printf("Enter element a%d%d: ", i + 1, j + 1);
        scanf("%d", &a[i][j]);
    }
// printing the matrix a[][]
printf("\nEnter matrix: \n");
for (int i = 0; i < r; ++i)
    for (int j = 0; j < c; ++j) {
        printf("%d ", a[i][j]);
    }
    printf("\n");
}

// computing the transpose
for (int i = 0; i < r; ++i)
    for (int j = 0; j < c; ++j) {
        transpose[j][i] = a[i][j];
    }
// printing the transpose
printf("\nTranspose of the matrix:\n");
for (int i = 0; i < c; ++i)
    for (int j = 0; j < r; ++j) {
        printf("%d ", transpose[i][j]);
    }
    printf("\n");
}
return 0;
}

```

Session -7
Creating and Running C Programs on User Defined Functions:

- 1. Write a C program to read a number, find its factorial using recursive function.**

```
#include<stdio.h>
int factorial(int n)
{
    if (n>=1)
        return n*factorial(n-1);
    else
        return 1;
}

int main()
{
    int n, fact;
    printf("Enter a positive integer: ");
    scanf("%d",&n);
    fact = factorial(n);
    printf("Factorial of %d = %d", n, fact );
    return 0;
}
```

- 2. Write a C program to read a number, find whether its prime number using functions (all categories).**

```
#include<stdio.h>

int check_prime(int);

main()
{
    int n, result;

    printf("Enter an integer to check whether it is prime or not.\n");

    scanf("%d",&n);

    result = check_prime(n);

    if (result == 1)
        printf("%d is prime.\n", n);

    else

        printf("%d is not prime.\n", n);
}
```

```

        return 0;
    }

    int check_prime(int a)
    {
        int c;
        for ( c = 2 ; c <= a - 1 ; c++ )
        {
            if ( a%c == 0 )
                return 0;
        }
        return 1;
    }
}

```

3. Write a program to find the smallest and largest element of an array using functions.

```

#include<stdio.h>
int largest(int a[], int n);
int smallest(int a[], int n);

int main()
{
    int arr[50],i,n;
    printf("\n Enter the array size:");
    scanf("%d",&n);
    printf("Enter the Array:");
    for(i=0;i<n;i++)
        scanf("%d",&arr[i]);
    smallest(arr, n);
    largest(arr,n);
    return 0;
}

int smallest (int a[], int n)
{
    int i;
    int small = a[0];
    for(i=1;i<n;i++)
    {
        if(a[i]<small)
            small=a[i];
    }
    printf("\nThe smallest element is %d",small);
}

```

```

int largest(int a[], int n)
{
    int i;
    int large = a[0];
    for(i=1;i<n;i++)
    {
        if(a[i]>large)
            large=a[i];
    }
    printf("\nThe largest element is %d",large);
}

```

4. **Write a program using functions to swap two integer values using call by reference.**

```

#include<stdio.h>
void swap_call_by_val(int, int);
void swap_call_by_ref(int *,int *);
int main()
{
    int a=1, b=2, c=3, d=4;
    printf("\n in main(), a = %d and b = %d",a,b);
    swap_call_by_val(a,b);
    printf("\n in main(), a = %d and b = %d",a,b);
    printf("\n\n in main(), c = %d and d = %d",c,d);
    swap_call_by_ref(&c,&d);
    printf("\n\n in main(), c = %d and d = %d",c,d);
    return 0;
}

void swap_call_by_val(int a, int b)
{
    int temp;
    temp = a;
    a = b;
    b = temp;
    printf("\n In function (call By Value Method) a = %d and b = %d", a,b);
}

void swap_call_by_ref(int *c, int *d)
{
    int temp;
    temp = *c;
    /*operator used to refer to the value
    *c = *d;
    *d = temp;
    printf("\n in function (call by reference Method) c = %d and d= %d", *c, *d);
}

```

Session -7

Creating and Running C Programs on Strings:

1. C program to read two strings, compare them without using C built-in functions.

```
#include<stdio.h>

#include<string.h>

int main()
{
    char str1[100], str2[100];

    int i=0, len1=0, len2=0, same=0;

    printf("\n enter the first string : ");

    gets(str1);

    printf("\n enter the second string : ");

    gets(str2);

    len1 = strlen(str1);

    len2 = strlen(str2);

    if(len1 == len2)
    {
        while(i<len1)
        {
            if(str1[i] == str2[i])
                i++;

            else break;
        }

        if(i==len1)
        {
            same=1;

            printf("\n The two strings are equal");
        }
    }
}
```

```

if(len1 !=len2)

    printf("\n The two strings are not equal");

if(same == 0)
{
    if(str1[i]>str2[i])

        printf("\n String1 is greater than String2");

    else if(str1[i]<str2[i])

        printf("\n String2 is greater tahn string1 ");

    return 0;
}
}

```

2.C program to read two strings, concatenating them without using string built-in functions.

```

#include<stdio.h>

#include<string.h>

int main()

{

    char str1[100], str2[100], str3[100];

    int i=0, j=0;

    printf("\n enter the first string : ");

    gets(str1);

    printf("\n enter the second string : ");

    gets(str2);

    while(str1[i] != '\0')

    {

        str3[i] = str1[i];

        i++;
    }
}

```



```

}

j=0;

while(str2[j] != '\0')

{

    str3[i] = str2[j];

    i++;

    j++;

}

str3[i] = '\0';

printf("\n The concatenated string is:");

puts(str3);

return 0;

}

```

3. Write a program using built-in string function to

a. To convert "good morning" to "GOOD MORNING".

```
#include <ctype.h>
```

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

```
int main()
```

```
{
```

```
    int j = 0;
```

```
    char str[] = "good morning";
```

```
    char ch;
```

```
    while (str[j]) {
```

```
        ch = str[j];
```

```
        putchar(toupper(ch));
```

```
        j++;
```

```
}
```

```
return 0;
```

Output: GOOD MORNING

b. Count the number of characters in “good Morning”.

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

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

```
int main()
```

```
{
```

```
    char str[] = "GOOD MORNING";
```

```
    printf("\n Length of the string is : %d",strlen(str));
```

```
    return 0;
```

```
}
```

c. To append the word “All” to the string “Good Morning”.

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

```
int main()
```

```
{
```

```
    char str1[50]= "Good Morning";
```

```
    char str2[50]=" All";
```

```
    strncat(str1, str2, 4);
```

```
    printf("\n str1: %s", str1);
```

```
    return 0;
```

```
}
```

d. reverse the string “Morning ” and check if the given string is a palindrome or not.

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

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

```
int main()
{
    char str[100] = "morning", b[100], temp;

    int j,i=0;

    strcpy(b,str); /* Copying input string */

    j = strlen(str)-1;

    while(i<j)
    { temp = str[j];

      str[j] = str[i];

      str[i] = temp;

      i++;

      j--;

    }

    if (strcmp(str, b) == 0) /* Comparing input string with the reverse string */

        printf("The string is a palindrome\n");

    else

        printf("the reverse string is %s\n",str);

    printf("The string is not a palindrome\n");

    return 0;

}
```

Session -9
Creating and Running C Programs on Storage Classes and Pointers:

1. C-program to show the use of auto and static variable.

```
#include <stdio.h>
void fun(void)
{
    auto int a=0;
    static int b=0;
    printf("a = %d, b = %d\n",a,b);
    a++;
    b++;
}

int main()
{
    int loop;
    //calling function 10 times
    for(loop=0; loop<5; loop++)
        fun();
    return 0;
}
```

2. C-program to add elements of an array using pointers.

```
#include <stdio.h>
void main(){
    int i, n, sum = 0, int a[20];
    int *ptr;
    ptr = a;
    printf("Enter size of array : ");
    scanf("%d", &n);
    printf("Enter elements in the List\n ");
    for (i = 0; i < n; i++){
        scanf("%d", &a[i]);
    }
    //calculate sum of elements
    for (i = 0; i < n; i++){
        sum = sum + *(ptr + i);
    }
    printf("Sum of all elements in an array is = %d", sum);
}
```

3. C-program to swap two numbers using pointers.

```
#include <stdio.h>
int main()
{
    int x, y, *a, *b, temp;
    printf("Enter the value of x and y\n");
    scanf("%d%d", &x, &y);
    printf("Before Swapping\nx = %d\ny = %d\n", x, y);
    a = &x;
    b = &y;
    temp = *b;
    *b = *a;
    *a = temp;
    printf("After Swapping\nx = %d\ny = %d\n", x, y);
    return 0;
}
```

Session -10
Creating and Running C Programs on Structures:

1. Write a C program using structures to read and display the information about an employee.

```
#include <stdio.h>
/*structure declaration*/
struct employee
{
    char   name[30];
    int    empId;
    float  salary;
};

int main()
{
    /*declare structure variable*/
    struct employee emp;
    /*read employee details*/
    printf("\nEmployee details :\n");
    printf("Enter the Name of an employee:");
    gets(emp.name);
    printf("Enter the employee ID:");
    scanf("%d",&emp.empId);
    printf("Enter the Salary of an employee:");
    scanf("%f",&emp.salary);
    /*print employee details*/
    printf("\nEntered detail is:");
    printf("Name: %s\n",emp.name);
    printf("Id: %d\n",emp.empId);
    printf("Salary: %f\n",emp.salary);
    return 0;
}
```

2. Define a structure containing the following details for 5 students: Name of Student, Age, Marks of 5 subjects [Use array of Structures]. Generate a report for the following
 - a. The total average marks for each student
 - b. The class average for each subject

```
#include<stdio.h>
struct student
{
    char sname[50];
    int roll;
    int m1,m2,m3,m4,m5;
} s[5];

int main()
{
    int i;
    float avgmarks[5], classavg_sub1=0, classavg_sub2=0, classavg_sub3=0,
classavg_sub4=0, classavg_sub5=0;
    printf("Enter information of 5 students:\n");
    // storing information
    for (i = 0; i < 5; ++i)
    {
        s[i].roll = i + 1;
        printf("Enter the roll number\n");
        scanf("%d", &s[i].roll);
        printf("Enter student name:");
        scanf("%s", s[i].sname);
        printf("Enter marks of all 5 subjects:\n ");
        scanf("%d%d%d%d%d", &s[i].m1,&s[i].m2,&s[i].m3,&s[i].m4,&s[i].m5);
    }
    // Displaying information

    printf("*****STUDENT DETAILS ARE:*****");
    for (i = 0; i < 5; ++i)
    {
        printf("\nRoll number: %d",s[i].roll);
        printf("\nstudent name:");
        puts(s[i].sname);
        printf("m1=%d\tm2=%d\tm3=%d\tm4=%d\tm5=%d\t\n",
s[i].m1,s[i].m2,s[i].m3,s[i].m4,s[i].m5);
        //Average marks of each student
        avgmarks[i]=(s[i].m1+s[i].m2+s[i].m3+s[i].m4+s[i].m5)/5;
        // class average of each subject
```

```

classavg_sub1=classavg_sub1+s[i].m1;
classavg_sub2=classavg_sub2+s[i].m2;
classavg_sub3=classavg_sub3+s[i].m3;
classavg_sub4=classavg_sub4+s[i].m4;
classavg_sub5=classavg_sub5+s[i].m5;
printf("Average marks of student-%d is=%f\n",s[i].roll,avgmarks[i]);
}

// class average of each subject
printf("*****CLASS AVERAGE OF EACH SUBJECT*****\n");
printf("class average of subject-1=%f\n",classavg_sub1/5);
printf("class average of subject-2=%f\n",classavg_sub2/5);
printf("class average of subject-3=%f\n",classavg_sub3/5);
printf("class average of subject-4=%f\n",classavg_sub4/5);
printf("class average of subject-5=%f\n",classavg_sub5/5);
return 0;
}

```