C LAB MANUAL R23

Week1/LAB1:

1. Basic Linux environments and its editors like vi, vim, emacs etc.

What is Linux or Unix environment?

Unix was originally decided to run on large, expensive mainframe computers. While Linux is graphical or text-based user interface of Unix isgraphical text-based command line interface of Linux is Bash, Zsh, Tesh.

What are the types of c program executers in Unix.

1. Use the vim, editor.
2. Vim.file.c (file name can be anything, but it should end with dot c extensioncommand).
3. Press I or insert button to go to insert made.
4. Press esc button and then type 'wq'. It will save the file.
5. gcc.file.c
6. ./a.out
7. In the file tab click new.
8. In execute tab.

What is Linux operating system used for?

Linux is used in the following says serve OS for web series data baseservers, file servers, email servers and any other type of shared server. Design to support high volume and multithreading applications. Linux is well suited for all types of serverapplications desktop for personal productivity computing.

2. Exposure to turbo C, gcc.

What is turbo c in program?

Turbo C is an integrated development environment (IDE) Created by boren software corporation 1990 it was one of the most popular computers musicduring the 1990s and early 2000's for the development of applications.

Below are the steps to run program in turbo C

1. type the program.
2. Solve the program with proper file extension C for C type files.
3. Compile the code by pressing Alt +F9.
4. If any errors are there edit the file save and recompile it till you get zero errors.

5. Run the program by pressing control + F9.
6. To see the output window press F5.

/*2:**Simple programs using scan() and printf() Print “hello world”*/**

```
#include<stdio.h>

int main()

{

Printf(“Hello world”);

Return 0;

}
```

OUTPUT:

Hello world

/* 3:Addition of 2 numbers*/

```
#include<stdio.h>

Int main()

{

Int a,b,sum;

Printf(“Enter two values”);

Scanf(“%d%d”,&a,&b);

Sum=a+b;

Printf(“ sum of two numbers is %d\t”,sum);

Return 0;

}
```

INPUT:

Enter two values

6

4

OUTPUT:

Sum of two number is 10

Week2/Lab2:

/*Algorithm to find sum and average of three numbers*/

Step 1: Start

Step 2 : Read the three number suppose a,b,c form the user.

Step 3: Declared a variable sum and Avg.

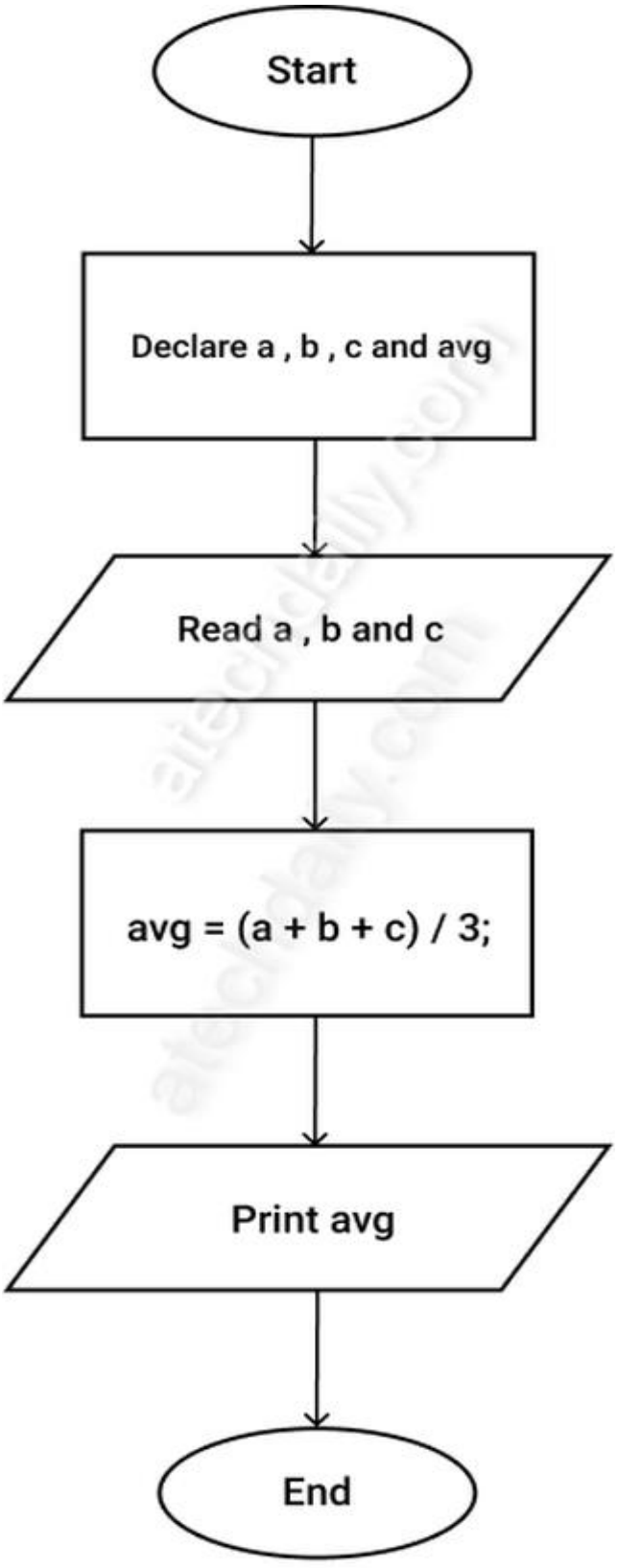
Step 4 : sum=a+b+c;

Step 5: Avg=sum/3

Step 6:Display sum "and Avg.

Step 7 : End .

Flow Chart:



/* 1:C Program to Find Sum and Average of 3 Numbers*/

```
#include <stdio.h>

int main()

{

   int a, b, c, sum;

   float avg;

   printf("Enter 3 numbers: \n");

   scanf("%d %d %d", &a, &b, &c);

      sum = a + b + c;

    avg = sum / 3;

   printf("Sum = %d \n", sum);

   printf("Average = %.2f", avg);

   return 0;

}
```

INPUT:

Enter 3 numbers:

6

8

4

OUTPUT

Sum = 18

Average = 6.00

/*2:algorithm to convert celsious to Fahrenheit*/

Step 1 : start

Step 2 : read the value of temperature in celisious say cel

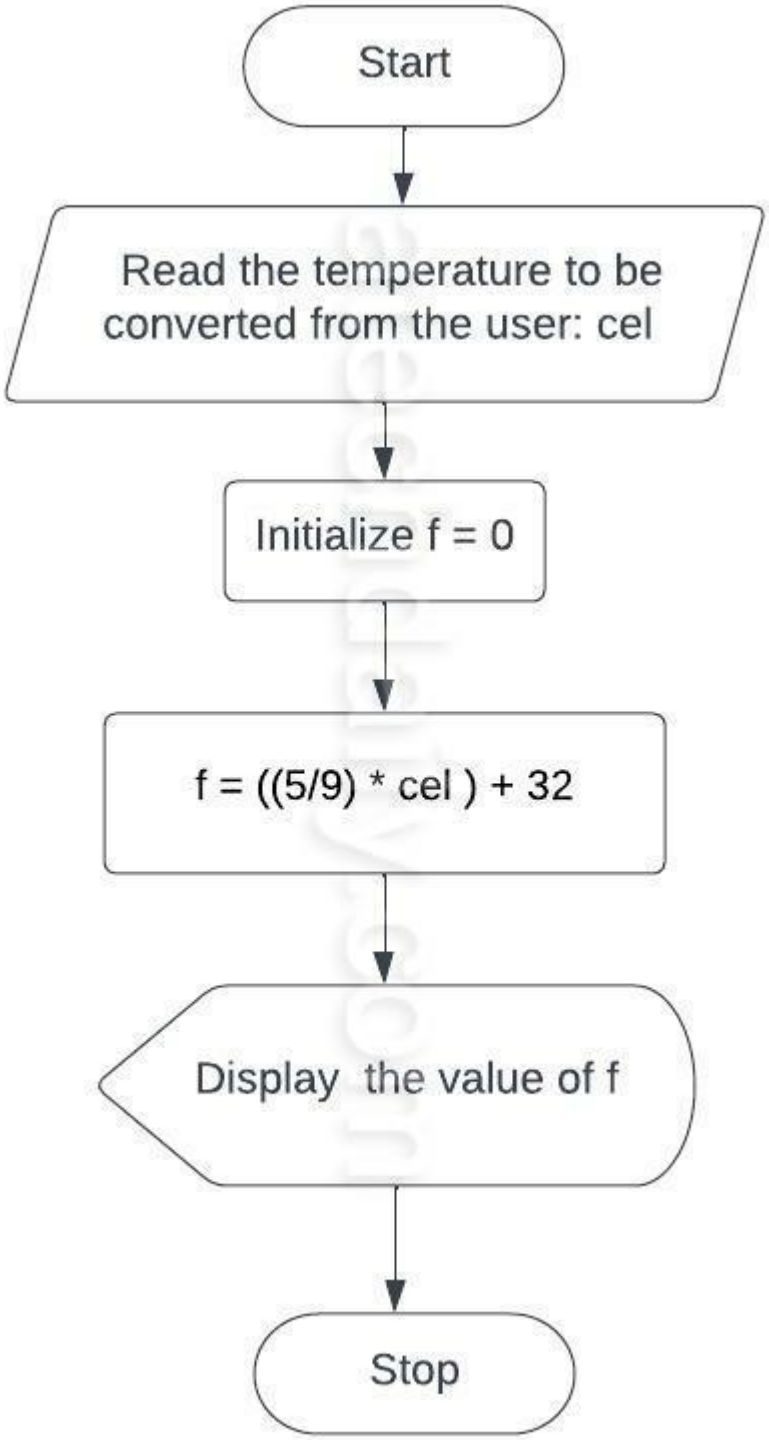
Step 3 : Initialize f=0

Step 4 : f=((5/9)*cel)+32

Step 5 : Display f

Step 6 : Stop

/*Flow chart to convert celsious to Fahrenheit**/



/*3:C Program to convert celsious to Fahrenheit */

```
#include <stdio.h>

int main()

{

   float celsius, fahrenheit;

   printf("Enter temperature in Celsius: ");

   scanf("%f", &celsius);

   //celsius to fahrenheit conversion formula

   fahrenheit = (celsius * 9 / 5) + 32;

   printf("%.2f Celsius = %.2f Fahrenheit", celsius, fahrenheit);

   return 0;

}
```

INPUT:

Enter temperature in Celsius: 10

OUTPUT:

10.00 Celsius = 50.00 Fahrenheit

/*3:algorithm to convert Fahrenheit to celsious*/

Step 1 : start

Step 2 : read the value of temperature in Fahrenheit say f

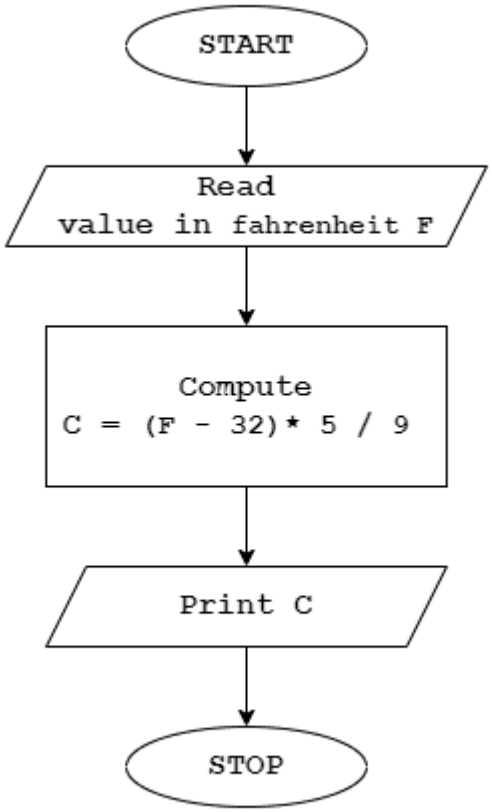
Step 3 : Initialize c=0

Step 4 : c=(f-32)*5/9

Step 5 : Display c

Step 6 : Stop

Flowchart to convert fahrenheit to celsius



https://www.alphabetacoder.com

/* 3: C Program to convert fahrenheit to celsious*/

```
#include<stdio.h>

int main(){

  float fahrenheit, celsius;

  //get the limit of fibonacci series

  printf("Enter Fahrenheit: ");

  scanf("%f",&fahrenheit);

  celsius = (fahrenheit - 32)*5/9;

  printf("Celsius: %f ", celsius);

  return 0;

}
```

INPUT:

Enter Fahrenheit: 100

OUTPUT:

Celsius: 37.777779

/*Algorithm to find simple Interest*/

Step 1: Start.

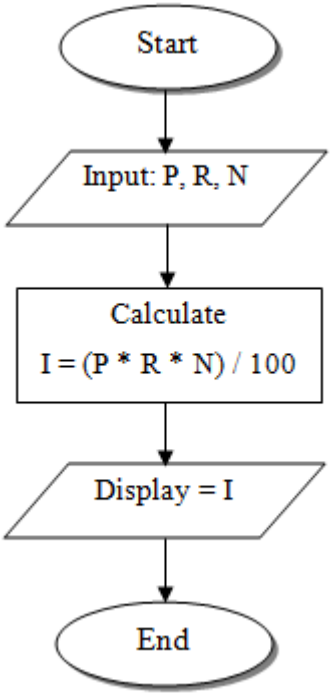
Step 2 : Read P,R, N.

Step 3 : SI=(P*R*N)/100.

Step 4 : Print SI

Step 5 : Stop

Flow chart to calculate simple interest



/*4:C program to calculate simple interest*/

```
#include <stdio.h>

int main()

{

   float principle, time, rate, SI;

   printf("Enter principle (amount): ");

   scanf("%f", &principle);

   printf("Enter time: ");

   scanf("%f", &time);

   printf("Enter rate: ");

   scanf("%f", &rate);

   /* Calculate simple interest */

   SI = (principle * time * rate) / 100;

   printf("Simple Interest = %f", SI);

   return 0;

}
```

INPUT:

```
Enter principle (amount): 1200

Enter time: 2

Enter rate: 5.4
```

OUTPUT:

```
Simple Interest = 129.600006
```

Week3/LAB3:

/* 1:C program to find the square root of a given number using sqrt()*/

```
#include <stdio.h>

int main()

{

   double n;

   printf("Enter number: ");

   scanf("%lf",&n);

     double square_root=sqrt(n);

    //Precision upto 2 decimal places

   printf("Number is %0.2lf and its square root is: %0.2lf",n,sqrt(n));

    return 0;

}
```

INPUT:

```
Enter number: 25
```

OUTPUT:

```
Number is 25.00 and its square root is: 5.00
```

/*2:c program to calculate compound interest*/

```
#include <stdio.h>
#include <math.h>
int main()
{
   float p, r, t, ci;
   printf("Enter the principle :");
   scanf("%f", &p);
   printf("Enter the rate :");
   scanf("%f", &r);
   printf("Enter the time :");
   scanf("%f", &t);
   ci = p * pow((1 + r / 100), t) - p;
   printf("\nThe compound interest is %0.2f", ci);
   return 0;
}
```

INPUT:

```
Enter the principle :5000
Enter the rate :2
Enter the time :3
```

OUTPUT:

```
The compound interest is 306.04
```

/*3:c program to calculate area of the triangle busing herons formula*/

```
# include<stdio.h>
# include <math.h>
int main()
{
   float a,b,c,s,area;
   printf("Enter the length of three sides of a triangle:\n");
   scanf("%f%f%f", &a,&b,&c);
   s = (a+b+c)/2;
   printf("The value of S is %.2f\n",s);
   area = sqrt(s*(s-a) * (s-b) * (s-c));
   printf("The area of the triangle is:%f\n", area);
   return 0;
}
```

INPUT:

```
Enter the length of three sides of a triangle:
6
8
12
```

OUTPUT:

```
The value of S is 13.00
The area of the triangle is:21.330729
```

/*4:c program to calculate distance travelled by an object*/

```
#include<stdio.h>

int main() {
  float u, a, d;
  int t;

  printf("\nEnter the value of a : ");
  scanf("%f", & a);

  printf("\nEnter the value of u : ");
  scanf("%f", & u);

  printf("\nEnter the value of t : ");
  scanf("%d", & t);

  d = (u * t) + (a * t * t) / 2;

  printf("\n The Distance : %.2f", d);

  return 0;
}
```

INPUT:

Enter the value of a : 8

Enter the value of u : 7

Enter the value of t : 6

OUTPUT:

The Distance : 186.00

Week4/Lab4:

/*1:C program to find maximum of three numbers using conditional operator.*/

```
#include <stdio.h>
 int main()
{
   int a, b, c, max;
   printf("Enter Three Integers\n");
   scanf("%d %d %d", &a, &b, &c);
   max = (a > b) ? ((a > c) ? a : c) : ((b > c) ? b : c);
   printf("Maximum Number is = %d\n", max);
   return 0;
}
```

INPUT:

Enter Three Integers

1

2

3

OUTPUT:

Maximum Number is = 3

/*2:write a c program to take marks of 5 subjects in integers and find the total and average in float*/

```
#include<stdio.h>

int main()

{

int chem,phy,eng,maths,bio;

float total,average;

printf("Enter the marks of five subjects: \n");

scanf("%d%d%d%d%d", &chem, &phy, &eng, &maths, &bio);

total= chem + phy + eng + maths + bio;

average = total / 5;

printf("Total marks = %.2f \n", total);

printf("Average marks = %.2f \n", average);

return 0;

}
```

INPUT:

```
Enter the marks of five subjects:

78

67

99

55

44
```

OUTPUT:

```
Total marks = 343.00

Average marks = 68.60
```

/*3:Evaluate the following expressions.

a. A+B*C+(D*E) + F*G

b. A/B*C-B+A*D/3

c. A+++B---A

d. J= (i++) + (++i)*/

```
#include <stdio.h>
int main()
{
   int A = 5, B = 3, C = 2, D = 7, E = 4, F = 6, G = 8;
   int i = 10, J;
   int result_a = A + B * C + (D * E) + F * G;
   printf("a. A+B*C+(D*E) + F*G = %d\n", result_a);
   int result_b = A / B * C - B + A * D / 3;
   printf("b. A/B*C-B+A*D/3 = %d\n", result_b);
   int result_c = A++ + ++B - A;
   printf("c. A+++B---A = %d\n", result_c);
   J = (i++) + (++i);
   printf("d. J = (i++) + (++i) = %d\n", J);
   return 0;
}
```

OUTPUT:

```
a. A+B*C+(D*E) + F*G = 87
b. A/B*C-B+A*D/3 = 10
c. A+++B---A = 3
d. J = (i++) + (++i) = 22
```

WEEK5/LAB5:

/*1:c program to find maximum of four numbers*/

```
#include <stdio.h>
int main()
{
   int a, b, c, d;
   printf("enter four numbers:");
   scanf("%d %d %d %d", &a, &b, &c, &d);
   if (a > b)
   {
      if (a > c)
      {
         if (a > d)
         {
            printf("%d is greatest \n", a);
         }
      }
   }
   else
   {
      if (b > c)
      {
         if (b > d)
         {
            printf("%d is greatest \n", b);
         }
```

```
    }
    else
    {
       if (c > d)
       {
          printf("%d is greatest \n", c);
       }
       else
       {
          printf("%d is greatest \n", d);
       }
    }
  }
```

INPUT:

enter four numbers:2

4

6

8

OUTPUT:

8 is greatest

/*2:c program to generate electricity bill*/

```
#include <stdio.h>
#define UNIT_RATE 7.5  // Rate per unit consumed
#define TAX_RATE 0.1   // Tax rate
  //Function to calculate the bill amount
float calculateBill(int units) {
    float billAmount, taxAmount;
     // Calculate the bill amount
   billAmount = units * UNIT_RATE;
      // Calculate the tax amount
    taxAmount = billAmount * TAX_RATE;
      // Add tax to the bill amount
    billAmount += taxAmount;
      return billAmount;
}

int main()
 {
  int units;
    float totalBill;
      // Prompt the user to enter units consumed
   printf("Enter the number of units consumed: ");
    scanf("%d", &units);
   // Calculate the bill amount
   totalBill = calculateBill(units);

    // Display the bill amount
    printf("Electricity Bill Amount: $%.2f\n", totalBill);

    return 0;
}
```

INPUT:

Enter the number of units consumed: 800

OUTPUT:

Electricity Bill Amount: $6600.00

/*3:c program to find roots of quadratic equation*/

```
#include <stdio.h>
#include <math.h>
void main()
{
   int a, b, c, d;      float x1, x2;
   printf("Input the value of a, b & c : ");
   scanf("%d%d%d", &a, &b, &c);      d = b*b - 4*a*c;
   if(d == 0)
{
     printf("Both roots are equal.\n");
     x1 = -b / (2.0 * a);
     x2 = x1;
     printf("First  Root Root1 = %f\n", x1);
     printf("Second Root Root2 = %f\n", x2);
   }
   else if(d > 0)
   {
       printf("Both roots are real and different.\n");
       x1 = (-b + sqrt(d)) / (2 * a);
       x2 = (-b - sqrt(d)) / (2 * a);
       printf("First  Root Root1 = %f\n", x1);
       printf("Second Root Root2 = %f\n", x2);
   }
   else
       printf("Roots are imaginary;\nNo Solution. \n");
}
```

INPUT:

Input the value of a, b & c : 1

8

2

OUTPUT:

Both roots are real and different.

First Root Root1 = -0.258343

Second Root Root2 = -7.741657

INPUT:

Input the value of a, b & c : 1

5

7

Roots are imaginary;

No Solution.

INPUT:

Input the value of a, b & c : 1

2

1

OUTPUT:

Both roots are equal.

First Root Root1 = -1.000000

Second Root Root2 = -1.000000

/*4:c program to generate a simple calculator program using switch..case*/

```
#include <stdio.h>   // Include the standard input/output header file.
#include <stdio.h>
int main() {
  char op;
  double first, second;
  printf("Enter an operator (+, -, *, /): ");
  scanf("%c", &op);
  printf("Enter two operands: ");
  scanf("%lf %lf", &first, &second);
  switch (op) {
    case '+':
      printf("%.1lf + %.1lf = %.1lf", first, second, first + second);
      break;
    case '-':
      printf("%.1lf - %.1lf = %.1lf", first, second, first - second);
      break;
    case '*':
      printf("%.1lf * %.1lf = %.1lf", first, second, first * second);
      break;
    case '/':
      printf("%.1lf / %.1lf = %.1lf", first, second, first / second);
      break;
       default:
      printf("Error! operator is not correct");
  }
```

```
  return 0;

}
```

INPUT:

Enter an operator (+, -, *, /): +

Enter two operands: 2

2

OUTPUT:

2.0 + 2.0 = 4.0

/*5:c program to find the given year is leap year or not*/

```
#include <stdio.h>

int main() {

  int year;

  printf("Enter a year: ");

  scanf("%d", &year);

  if (year % 400 == 0) {

    printf("%d is a leap year.", year);

  }

  else if (year % 100 == 0) {

    printf("%d is not a leap year.", year);

  }

  else if (year % 4 == 0) {

    printf("%d is a leap year.", year);

  }

  else {

    printf("%d is not a leap year.", year);

  }


  return 0;

}
```

INPUT:

Enter a year: 2020

OUTPUT:

2020 is a leap year.

INPUT:

Enter a year: 2023

OUTPUT:

2023 is not a leap year.

WEEL6/LAB6:

/*1:C program to find the factorial of a given number*/

```
#include<stdio.h>

int main()

{

 int i,fact=1,number;

 printf("Enter a number: ");

  scanf("%d",&number);

    for(i=1;i<=number;i++){

      fact=fact*i;

  }

  printf("Factorial of %d is: %d",number,fact);

return 0;

}
```

INPUT:

Enter a number: 5

OUTPUT:

Factorial of 5 is: 120

/*2:C program to find given number prime or not*/

```
#include <stdio.h>


int main() {
  int n,count=0,i;
  printf("Enter the number");
  scanf("%d",&n);
  for(i=1;i<=n;i++)
  {
     if(n%i==0)
     {
        count++;
     }
  }
     if(count==2)
     printf("Given number is prime number");
     else
     printf("given number is not prime");
    return 0;
  }
```

INPUT:

Enter the number7

OUTPUT:

Given number is prime number

INPUT:

Enter the number8

OUTPUT:

given number is not prime

/*3:C program to compute sine and cos series*/

```
#include<stdio.h>
#include <math.h>
#define PI 3.1416
#define MAX 150
main ( ) {
  int angle;
  float x,y;
  angle = 0;
  printf("Angle cos(angle)");
  while(angle <= MAX) {
    x = (PI/MAX)*angle;
    y = cos(x);
    printf("%15d %13.4f", angle, y);
    angle = angle + 10;
  }
}
```

OUTPUT:

```
Angle cos(angle)          0       1.0000          10       0.9781          20       0.9135          30
0.8090          40       0.6691          50       0.5000          60       0.3090          70       0.1045
80      -0.1045          90      -0.3090         100      -0.5000         110      -0.6691         120      -
0.8090         130      -0.9135         140      -0.9781         150      -1.0000
```

/*4:C program to check wether the given number is palindrome or not*/

```
#include<stdio.h>

int main()

{

int n,r,sum=0,temp;

printf("enter the number=");

scanf("%d",&n);

temp=n;

while(n>0)

{

r=n%10;

sum=(sum*10)+r;

n=n/10;

}

if(temp==sum)

printf("palindrome number ");

else

printf("not palindrome");

return 0;

}
```

INPUT:

```
enter the number=151
```

OUTPUT:

palindrome number

/*5:C program to construct pyramid of numbers*/

```
#include <stdio.h>
void main()
{     int i, j, rows;
   printf (" Enter a number of rows: \n ");
   scanf("%d", &rows);
   printf("\n");
   for (i = 1; i <= rows; ++i) // outer loop
   {
      for (j = 1; j <= i; ++j) // inner loop
      {
         printf ("* "); // print the Star
      }
      printf ("\n");
   }
}
```

INPUT:

Enter a number of rows:

5

OUTPUT:

```
 *
* *
* * *
* * * *
* * * * *
```

WEEK7/LAB7:

/*1:find the max and min of the1D integer array*/

```
#include <stdio.h>

void main()
{
    int arr1[100];
    int i, mx, mn, n;


     printf("\n\nFind maximum and minimum element in an array :\n");
     printf("-------------------------------------------------\n");
     printf("Input the number of elements to be stored in the array :");
     scanf("%d",&n);
     printf("Input %d elements in the array :\n",n);
     for(i=0;i<n;i++)
       {
           printf("element - %d : ",i);
           scanf("%d",&arr1[i]);
         }


    mx = arr1[0];
    mn = arr1[0];

    for(i=1; i<n; i++)
    {
        if(arr1[i]>mx)
        {
            mx = arr1[i];
        }


        if(arr1[i]<mn)
        {
            mn = arr1[i];
        }
    }
    printf("Maximum element is : %d\n", mx);
    printf("Minimum element is : %d\n\n", mn);
}
```

INPUT:

Find maximum and minimum element in an array :

Input the number of elements to be stored in the array :4

Input 4 elements in the array :

element - 0 : 1

element - 1 : 2

element - 2 : 3

element - 3 : 4

OUTPUT:

Maximum element is : 4

Minimum element is : 1

/*2:C Program for linear search*/

```
#include <stdio.h>

int main()

{

  int array[100], search, c, n;

  printf("Enter number of elements in array\n");

  scanf("%d", &n);

  printf("Enter %d integer(s)\n", n);

  for (c = 0; c < n; c++)

    scanf("%d", &array[c]);

  printf("Enter a number to search\n");

  scanf("%d", &search);

  for (c = 0; c < n; c++)

  {

    if (array[c] == search)    /* If required element is found */

    {

      printf("%d is present at location %d.\n", search, c+1);

      break;

    }

  }

  if (c == n)

    printf("%d isn't present in the array.\n", search);

  return 0;

}
```

OUTPUT:

Enter number of elements in array

5

Enter 5 integer(s)

11

22

33

44

55

Enter a number to search

44

44 isn't present in the array.

OUTPUT:

Enter number of elements in array

5

Enter 5 integer(s)

11

22

33

44

55

Enter a number to search

77

77 isn't present in the array.

/*3:c program for reverse of a given array*/

```
#include<stdio.h>
int main()
{
   int arr[10], i;
   printf("Enter any 5 elements: ");
   for(i=0; i<5; i++)
   {
      scanf("%d", &arr[i]);
   }
   printf("\nThe array elements in reverse order:\n");
   for(i=4; i>=0; i--)
   {
      //if(i==0)
         printf("%d", arr[i]);
      //else
        // printf("%d, ", arr[i]);
   }
      return 0;
}
```

INPUT:

Enter any 5 elements:

1

2

3

4

5

OUTPUT:

The array elements in reverse order:

5

4

3

2

1

/*4:C program for 2’s complement of a given binary number*/

```
#include <stdio.h>
int main()
{
  int n;  // variable declaration
  printf("Enter the number of bits do you want to enter :");
  scanf("%d",&n);
  char binary[n+1];  // binary array declaration;
  char onescomplement[n+1]; // onescomplement array declaration
  char twoscomplement[n+1]; // twoscomplement array declaration
  int carry=1; // variable initialization
  printf("\nEnter the binary number : ");
  scanf("%s", binary);
  printf("%s", binary);
  printf("\nThe ones complement of the binary number is :");

  // Finding onescomplement in C
  for(int i=0;i<n;i++)
  {
     if(binary[i]=='0')
     onescomplement[i]='1';
     else if(binary[i]=='1')
     onescomplement[i]='0';
  }
  onescomplement[n]='\0';
  printf("%s",onescomplement);
```

```
printf("\nThe twos complement of a binary number is : ");


// Finding twoscomplement in C
for(int i=n-1; i>=0; i--)
   {
      if(onescomplement[i] == '1' && carry == 1)
      {
         twoscomplement[i] = '0';
      }
      else if(onescomplement[i] == '0' && carry == 1)
      {
         twoscomplement[i] = '1';
         carry = 0;
      }
      else
      {
         twoscomplement[i] = onescomplement[i];
      }
   }
twoscomplement[n]='\0';
printf("%s",twoscomplement);
return 0;
}
```

OUTPUT:

Enter the number of bits do you want to enter :4

Enter the binary number : 1011

1011

The ones complement of the binary number is :0100

The twos complement of a binary number is : 0101

/*5:c program to remove the duplicate elements*/

```
#include <stdio.h>
int main()
{
   int n, count = 0;
   printf("Enter number of elements in the array: ");
   scanf("%d", &n);
   int arr[n], temp[n];
   if(n==0)
   {
      printf("No element inside the array.");
      exit(0);
   }
   printf("Enter elements in the array: ");
   for (int i = 0; i < n; i++)
   {
      scanf("%d", &arr[i]);
   }

   printf("\nArray Before Removing Duplicates: ");
   for (int i = 0; i < n; i++)
      printf("%d ", arr[i]);

   // To store unique elements in temp after removing the duplicate elements
   for (int i = 0; i < n; i++)
   {
```

```
        int j;
        for (j = 0; j < count; j++)
        {
          if (arr[i] == temp[j])
            break;
        }
        if (j == count)
        {
          temp[count] = arr[i];
          count++;
        }
    }
     printf("\nArray After  Removing Duplicates: ");
    for (int i = 0; i < count; i++)
        printf("%d ", temp[i]);

    return 0;
}
```

OUTPUT:

Enter number of elements in the array: 5

Enter elements in the array:

1

1

2

3

4

Array After Removing Duplicates

1

2

3

4

WEEK8/LAB 8:

/*1:C Program to find addition of two matrices*/

```
#include <stdio.h>
int main()
{
int a[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
int b[3][3] = {{9, 8, 7}, {6, 5, 4}, {3, 2, 1}};
int c[3][3];
int i, j;
for (i = 0; i < 3; i++)
 {
 for (j = 0; j < 3; j++)
 {
 c[i][j] = a[i][j] + b[i][j];
 } }
 printf("Result of addition: \n");
 for (i = 0; i < 3; i++)
 {
      for (j = 0; j < 3; j++)
 {
          printf("%d ", c[i][j]);
  }
  printf("\n");
  }
  return 0;
  }
```

OUTPUT:

Result of addition:

10 10 10

10 10 10

10 10 10

/*2:c program to find multiplication of 2 matrices*/

```
#include<stdio.h>
void main()
{
int a[3][3],b[3][3],c[3][3],i,j,k;
printf("\n Enter 1st matrix:");
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
scanf("%d",&a[i][j]);
}
}
printf("\n Enter 2nd matrix:");
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
scanf("%d",&b[i][j]);
}
}
printf("\n The multiplication of matrix is:\n");
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
```

```
c[i][j]=0;
for(k=0;k<3;k++)
{
c[i][j]+=a[i][k]*b[k][j];
}
printf("\t%d",c[i][j]);
}
printf("\n");
}
}
```

INPUT:

Enter 1st matrix:

1 2 3

1 4 7

1 5 9

Enter 2nd matrix:

5 7 9

5 4 3

9 8 7

OUTPUT:

The multiplication of matrix is:

42	39	36
88	79	70
111	99	87

/* 3:c program to Sort array elements using bubble sort */

```
#include <stdio.h>
void bubble_sort(int arr[], int n)
{
  int i, j;
  for (i = 0; i < n - 1; i++)
  {
    for (j = 0; j < n - i - 1; j++)
    {
      if (arr[j] > arr[j + 1])
      {
        int 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]);
  bubble_sort(arr, n);
  printf("Sorted array: ");
  for (int i = 0; i < n; i++)
{
    printf("%d ", arr[i]);
```

```
  }

  return 0;

}
```

OUTPUT:

Sorted array: 11 12 22 25 34 64 90

/*4:C program to Concatenate two strings without built-in function*/

```
#include<stdio.h>
void main(void)
{
 char str1[25],str2[25];
 int i=0,j=0;
 printf("\nEnter First String:");
 gets(str1);
 printf("\nEnter Second String:");
 gets(str2);
 while(str1[i]!='\0')
 i++;
 while(str2[j]!='\0')
 {
 str1[i]=str2[j];
 j++;
 i++;
 }
 str1[i]='\0';
 printf("\nConcatenated String is %s",str1);
}
```

OUTPUT:

Enter First String:hello

Enter Second String:world

Concatenated String is hello world

/*5:c program to peint Reverse a string using built-in and without built-in string functions*/

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

void main()
{
   int i,n;
   char str[20];
   printf("Enter the String to get reversed\n");
   gets(str);
   n=strlen(str);
   printf("\nReversed string is \n");
   for(i=n-1;i>=0;i--)
   {
     printf("%c",str[i]);
   }

}
```

OUTPUT:

```
 Enter the String to get reversed

hello good morning

Reversed string is

gninrom doog olleh
```

WEEK9/LAB9

/*1.Write a C program to find the sum of a 1D array using malloc()*/

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

int main()
{
	int i;
	int count;
	int *arr;
	int sum = 0;
	printf("Enter the total number of elements you want to enter : ");
	scanf("%d", &count);
	arr = (int *)malloc(count * sizeof(int));
	for (i = 0; i < count; i++)
	{

		printf("Enter element %d : ", (i + 1));
		scanf("%d", arr + i);


		sum += *(arr + i);
	}
	printf("sum is %d \n", sum);
	free(arr);
	return 0;
}
```

OUTPUT

```
Enter the total number of elements you want to enter : 5
Enter element 1 : 100
Enter element 2 : 200
Enter element 3 : 300
Enter element 4 : 400
Enter element 5 : 500
sum is 1500
```

/*2.write a C program to find the total,average of n students using structures*/

```
#include<stdio.h>
struct student
{
   char name[10];
   int rollno;
   int subject[5],total,average;
};
main ( )
{
   static struct student s[100];
   int n,i,j;
   printf("Enter the number of Students: ");
   scanf("%d",&n);
   printf("Enter the Marks of Five Subjects: ");
   for(i=0; i<n; i++)
   {
      printf("\nEnter student[%d] student marks",i);
      s[i].total=0;
      for(j=0; j<5; j++)
      {
         scanf("%d",&s[i].subject[j]);
         s[i].total=s[i].total+s[i].subject[j];
         s[i].average=s[i].total/5;
      }
      printf("sum=%d\n",s[i].total);
      printf("average=%d\n",s[i].average);
   }
}
```

OUTPUT

```
Enter the number of Students: 2
Enter the Marks of Five Subjects:
Enter student[0] student marks50 60 70 80 90
sum=350
average=70

Enter student[1] student marks45 67 78 59 79
sum=328
average=65
```

/*3.Enter n students data using calloc() and display failed students list*/

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

struct student {
   char name[50];
   int age;
   float marks;
};

int main() {
   int i, n, fail_count = 0;
   struct student *s;

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

   s = (struct student *) calloc(n, sizeof(struct student));

   for(i = 0; i < n; i++) {
      printf("Enter details of student %d:\n", i + 1);
      printf("Name: ");
      scanf("%s", (s + i)->name);
      printf("Age: ");
      scanf("%d", &(s + i)->age);
      printf("Marks: ");
      scanf("%f", &(s + i)->marks);

      if((s + i)->marks < 50) {
         fail_count++;
      }
   }

   printf("\nStudent details:\n");
   for(i = 0; i < n; i++) {
      printf("Name: %s\n", (s + i)->name);
      printf("Age: %d\n", (s + i)->age);
      printf("Marks: %.2f\n", (s + i)->marks);
   }

   printf("\nNumber of students who failed: %d\n", fail_count);

   free(s);
   return 0;
}
```

INPUT:

Enter the number of students: 3
Enter details of student 1:
Name: sai
Age: 19
Marks: 78
Enter details of student 2:
Name: ram
Age: 18
Marks: 36
Enter details of student 3:
Name: dev
Age: 19
Marks: 60

OUTPUT:

Student details:
Name: sai
Age: 19
Marks: 78.00
Name: ram
Age: 18
Marks: 36.00
Name: dev
Age: 19
Marks: 60.00

Number of students who failed: 1

/*4. Read student name and marks from the command line and display the student deatails a long with the total */

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

struct student {
    char name[50];
    float marks;
};

int main() {
    int i, n;
    float total_marks = 0;
    struct student *s;

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

    s = (struct student *) calloc(n, sizeof(struct student));

    for(i = 0; i < n; i++) {
        printf("Enter details of student %d:\n", i + 1);
        printf("Name: ");
        scanf("%s", (s + i)->name);
        printf("Marks: ");
        scanf("%f", &(s + i)->marks);

        total_marks += (s + i)->marks;
    }

    printf("\nStudent details:\n");
    for(i = 0; i < n; i++) {
        printf("Name: %s\n", (s + i)->name);
        printf("Marks: %.2f\n", (s + i)->marks);
    }

    printf("\nTotal marks: %.2f\n", total_marks);

    free(s);
    return 0;
}
```

INPUT:

Enter the number of students: 2
Enter details of student 1:
Name: geetha
Marks: 76
Enter details of student 2:
Name: jyothi
Marks: 86

OUTPUT:

Student details:
Name: geetha
Marks: 76.00
Name: jyothi
Marks: 86.00

Total marks: 162.00

/*5. write a c program to implement realloc*/

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

int main() {
   int i, n;
   int *ptr;

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

   ptr = (int*) malloc(n * sizeof(int));

   if(ptr == NULL) {
      printf("Memory allocation failed!");
      exit(0);
   }

   printf("Enter the elements:\n");
   for(i = 0; i < n; i++) {
      scanf("%d", ptr + i);
   }

   printf("Entered elements are:\n");
   for(i = 0; i < n; i++) {
      printf("%d ", *(ptr + i));
   }

   printf("\nEnter the new size: ");
   scanf("%d", &n);

   ptr = (int*) realloc(ptr, n * sizeof(int));

   printf("Enter the new elements:\n");
   for(i = 0; i < n; i++) {
      scanf("%d", ptr + i);
   }

   printf("New elements are:\n");
   for(i = 0; i < n; i++) {
      printf("%d ", *(ptr + i));
   }

   free(ptr);
   return 0;
}
```

OUTPUT

```
Enter the number of elements: 4
Enter the elements:
34
60
70
80
Entered elements are:
34 60 70 80
Enter the new size: 5
Enter the new elements:
40
70
80
90
50
New elements are:
40 70 80 90 50
```

WEEK 10/LAB10

/*1.create and display the single linked list using self-referential structure*/

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

struct node {
   int num;
   struct node *nextptr;
}*stnode;

void createNodeList(int n);
void displayList();

int main() {
   int n;
   printf("Linked List: To create and display Singly Linked List:\n");
   printf("-------------------------------------------------------------\n");
   printf("Input the number of nodes: ");
   scanf("%d", &n);
   createNodeList(n);
   printf("\nData entered in the list:\n");
   displayList();
   return 0;
}

void createNodeList(int n) {
   struct node *fnNode, *tmp;
   int num, i;
   stnode = (struct node*)malloc(sizeof(struct node));

   if(stnode == NULL) {
      printf("Memory can not be allocated.");
   } else {
      printf("Input data for node 1: ");
      scanf("%d", &num);
      stnode->num = num;
      stnode->nextptr = NULL;
      tmp = stnode;
```

```
        for(i = 2; i <= n; i++) {
            fnNode = (struct node*)malloc(sizeof(struct node));

            if(fnNode == NULL) {
                printf("Memory can not be allocated.");
                break;
            } else {
                printf("Input data for node %d: ", i);
                scanf("%d", &num);
                fnNode->num = num;
                fnNode->nextptr = NULL;
                tmp->nextptr = fnNode;
                tmp = tmp->nextptr;
            }
        }
    }
}

void displayList() {
    struct node *tmp;

    if(stnode == NULL) {
        printf("List is empty.");
    } else {
        tmp = stnode;

        while(tmp != NULL) {
            printf("Data = %d\n", tmp->num);
            tmp = tmp->nextptr;
        }
    }
}
```

OUTPUT:

```
Linked List: To create and display Singly Linked List:
------------------------------------------------------------
Input the number of nodes: 3
Input data for node 1: 78
Input data for node 2: 58
Input data for node 3: 48

Data entered in the list:
Data = 78
Data = 58
Data = 48
```

/*2.Demonstrate the difference between structures and unions using a C program*/

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

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

union employee {
   char name[50];
   int id;
   float salary;
};

int main() {
   struct student s1;
   union employee e1;

   printf("Size of structure = %lu bytes\n", sizeof(s1));
   printf("Size of union = %lu bytes\n", sizeof(e1));

   return 0;
}
```

OUTPUT:

```
Size of structure = 60 bytes
Size of union = 52 bytes
```

/*3.write a c program to shift/rotate using bitfields.*/

```
#include <stdio.h>

int main() {
    unsigned int num = 0x12345678;
    unsigned int shift = 4;

    printf("Original number: 0x%08X\n", num);

    // Left shift
    num = (num << shift) | (num >> (32 - shift));
    printf("Left shift by %d bits: 0x%08X\n", shift, num);

    // Right shift
    num = (num >> shift) | (num << (32 - shift));
    printf("Right shift by %d bits: 0x%08X\n", shift, num);

    return 0;
}
```

OUTPUT:

```
Left shift by 4 bits: 0x23456781
Right shift by 4 bits: 0x12345678
```

/*4.write a c program to copy one structure variable another structure of the same type.*/

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

struct person {
    char name[50];
    int age;
    float height;
};

int main() {
    struct person p1 = {"John", 25, 5.8};
    struct person p2;

    p2 = p1;

    printf("p1: Name=%s, Age=%d, Height=%.2f\n", p1.name, p1.age, p1.height);
    printf("p2: Name=%s, Age=%d, Height=%.2f\n", p2.name, p2.age, p2.height);

    return 0;
}
```

OUTPUT:

```
p1: Name=John, Age=25, Height=5.80
p2: Name=John, Age=25, Height=5.80
```

WEEK11 / LAB11

/*1.write a c function to calculate the ncr value*/

```
#include <stdio.h>
int fact(int n);
int nCr(int n, int r);

int main() {
   int n = 10, r = 5;
   printf("nCr(%d, %d) = %d\n", n, r, nCr(n, r));
   return 0;
}
int fact(int n) {
   int i, f = 1;
   for (i = 1; i <= n; i++) {
      f *= i;
   }
   return f;
}
int nCr(int n, int r) {
   return fact(n) / (fact(r) * fact(n - r));
}
```

OUTPUT:

```
nCr(10, 5) = 252
```

/*2.write a C function to find the length of the string*/

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

int main() {
   char str[100];
   int len;

   printf("Enter a string: ");
   scanf("%s", str);

   len = strlen(str);

   printf("Length of the string: %d\n", len);

   return 0;
}
```

OUTPUT:

```
Enter a string: welcome
Length of the string: 7
```

/*3.write a c function to transpose of a matrix.*/

```
#include <stdio.h>

void transpose(int matrix[][10], int rows, int cols) {
   int transposed[cols][rows];
   for (int i = 0; i < rows; ++i) {
      for (int j = 0; j < cols; ++j) {
         transposed[j][i] = matrix[i][j];
      }
   }
   printf("The transpose of the matrix is:\n");
   for (int i = 0; i < cols; ++i) {
      for (int j = 0; j < rows; ++j) {
         printf("%d ", transposed[i][j]);
      }
      printf("\n");
   }
}

int main() {
   int matrix[10][10], rows, cols;
   printf("Enter the number of rows and columns of the matrix: ");
   scanf("%d %d", &rows, &cols);
   printf("Enter the elements of the matrix:\n");
   for (int i = 0; i < rows; ++i) {
      for (int j = 0; j < cols; ++j) {
         scanf("%d", &matrix[i][j]);
      }
   }
   transpose(matrix, rows, cols);
   return 0;
}
```

OUTPUT:

```
Enter the number of rows and columns of the matrix: 2
2
Enter the elements of the matrix:
23 45
12 56
The transpose of the matrix is:
23 12
45 56
```

/*4.write a c function to demonstrate numerical integration of differential equations using Euler's methods*/

```
#include<stdio.h>
float fun(float x,float y)
{
   float f;
   f=x+y;
   return f;
}
main()
{
   float a,b,x,y,h,t,k;
   printf("\nEnter x0,y0,h,xn: ");
   scanf("%f%f%f%f",&a,&b,&h,&t);
   x=a;
   y=b;
   printf("\n  x\t  y\n");
   while(x<=t)
   {
      k=h*fun(x,y);
      y=y+k;
      x=x+h;
      printf("%0.3f\t%0.3f\n",x,y);
   }
}
```

OUTPUT:

```
Enter x0,y0,h,xn: 0 1 0.1 1

  x      y
0.100   1.100
0.200   1.220
0.300   1.362
0.400   1.528
0.500   1.721
0.600   1.943
0.700   2.197
0.800   2.487
0.900   2.816
1.000   3.187
```

WEEK 12 / LAB 12:

/*1:write a recursive function to generate Fibonacci series*/

```
#include<stdio.h>

void fibonacci(int);

int main()
{
   int limit;

   printf("Enter the number of terms to be printed\n");
   scanf("%d", &limit);

   fibonacci(limit);

   return 0;
}

void fibonacci(int num)
{
   int n1 = 0, n2 = 1, n3, count;

   printf("\nFibonacci Series ..\n");
   printf("1. %d\n2. %d\n", n1, n2);

   for(count = 3; count <= num; count++)
   {
      n3 = n1 + n2;
      printf("%d. %d\n", count, n3);

      n1 = n2;
      n2 = n3;
   }
}
```

OUTPUT:

```
Enter the number of terms to be printed
5

Fibonacci Series ..
1. 0
2. 1
3. 1
4. 2
5. 3
```

/*2:write a recursive function to find the lcm of two numbers*/

```
#include<stdio.h>
void fibonacci(int);
#include <stdio.h>
int lcm(int a, int b);

int main()
{
    int num1, num2, LCM;
   printf("Enter any two numbers to find lcm: ");
    scanf("%d%d", &num1, &num2);
     if(num1 > num2)
        LCM = lcm(num2, num1);
    else
        LCM = lcm(num1, num2);

    printf("LCM of %d and %d = %d", num1, num2, LCM);
      return 0;
}
int lcm(int a, int b)
{
    static int multiple = 0;
    multiple += b;
    if((multiple % a == 0) && (multiple % b == 0))
    {
        return multiple;
    }
    else
    {
        return lcm(a, b);
    }
}
```

Output:

Enter any two numbers to find lcm: 12 30

LCM of 12 and 30 = 60

/*3:write the recursive function to find the factorial of a number*/

```
#include<stdio.h>
long int factorial(int n);
int main() {
   int n;
   printf("Enter a positive integer: ");
   scanf("%d",&n);
   printf("Factorial of %d = %ld", n, factorial(n));
   return 0;
}

long int factorial(int n) {
   if (n>=1)
      return n*factorial(n-1);
   else
      return 1;
}
```

OUTPUT:

```
Enter a positive integer: 5
Factorial of 5 = 120
```

/*4:write c program to implement Ackermann function using recursion.*/

```
#include<stdio.h>
int A(int m, int n);

main()
{
	int m,n;
	printf("Enter two numbers :: \n");
	scanf("%d%d",&m,&n);
	printf("\nOUTPUT :: %d\n",A(m,n));
}

int A(int m, int n)
{
	if(m==0)
		return n+1;
	else if(n==0)
		return A(m-1,1);
	else
		return A(m-1,A(m,n-1));
}
```

OUTPUT::

1) Enter two numbers ::
   2 0

OUTPUT :: 3

2) Enter two numbers ::
   0 5

OUTPUT :: 6

/*5:write a recursive function to find the sum of series*/

```
#include <stdio.h>

int addNumbers(int n);

int main() {

  int num;
  printf("Enter a positive integer: ");
  scanf("%d", &num);
  printf("Sum = %d", addNumbers(num));
  return 0;
}

int addNumbers(int n) {
  if (n != 0)
    return n + addNumbers(n - 1);
  else
    return n;
}
```

OUTPUT:

```
Enter a positive integer: 6
Sum = 21
```

WEEK 13 / Lab 13:

/*1:write a c program to swap two numbers using call by reference*/

```
#include <stdio.h>
void swap(int * num1, int * num2);

int main()
{
   int num1, num2;
  printf("Enter two numbers: ");
   scanf("%d%d", &num1, &num2);
  printf("Before swapping in main n");
   printf("Value of num1 = %d \n", num1);
   printf("Value of num2 = %d \n\n", num2);
 swap(&num1, &num2);
 printf("After swapping in main n");
   printf("Value of num1 = %d \n", num1);
   printf("Value of num2 = %d \n\n", num2);
 return 0;
}
void swap(int * num1, int * num2)
{
   int temp;
 temp = *num1;
  *num1= *num2;
  *num2= temp;
   printf("After swapping in swap function n");
   printf("Value of num1 = %d \n", *num1);
   printf("Value of num2 = %d \n\n", *num2);
}
```

OUTPUT:

```
Enter two numbers: 10 20
Before swapping in main nValue of num1 = 10
Value of num2 = 20

After swapping in swap function nValue of num1 = 20
Value of num2 = 10

After swapping in main nValue of num1 = 20
Value of num2 = 10
```

/*2:Demonstrate dangling pointer problem using a c program*/

```
#include <stdio.h>
int main()
{
   int *ptr=(int *)malloc(sizeof(int));
   int a=560;
   ptr=&a;
   free(ptr);
   ptr=NULL
   return 0;
}
```

OUTPUT:

```
int a=560;
 free(): invalid pointer
```

/*3:Write a c program to copy one string to another using pointer*/

```
#include<stdio.h>
void copystr(char*,char*);
void main()
{
        char*str1="c programming lab";
        char str2[30];
         copystr(str2,str1);
        printf("\n %s",str2);
        getch();
 }
       void copystr(char *dest,char *src)
    {
          while(*src!='\0')
          *dest++=*src++;
          *dest='\0';
          return;
    }
```

OUTPUT:

```
c programming lab
```

/*4:write a c program to find no of lowercase,uppercase,digits and other charactersusing pointers.*/

```
#include <stdio.h>
int main()
{
    char inputString[100];
    int upperCount, lowerCount, specialCount, digitCount, i;
    printf("Enter a String : ");
    gets(inputString);
    printf("String input is %s ", inputString);
   upperCount = lowerCount = specialCount = digitCount = 0;
   for (i = 0; inputString[i] != '\0'; i++)
    {
        if (inputString[i] >= 'A' && inputString[i] <= 'Z')
        {
            upperCount++;
        }
        else if (inputString[i] >= 'a' && inputString[i] <= 'z')
        {
            lowerCount++;
        }
        else if (inputString[i] >= '0' && inputString[i] <= '9')
        {
            digitCount++;
        }
        else
        {
            specialCount++;
        }
    }
    printf("\nUpper case count : %d \n", upperCount);
    printf("Lower case count : %d \n", lowerCount);
    printf("Digit count : %d \n", digitCount);
    printf("Special character count : %d \n", specialCount);
  return 0;
}
```

OUTPUT:

```
Enter a String : Hello world 112@#$
String input is Hello world 112@#$
Upper case count : 1
Lower case count : 9
Digit count : 3
Special character count : 5
```

WEEK 14 / LAB 14

Write a C program to write and read text into a file

```
#include< stdio.h >
 int main()
 {
 FILE *fp; /* file pointer*/
 char fName[20];
printf("\nEnter file name to create :");
scanf("%s",fName); /*creating (open) a file*/
 fp=fopen(fName,"w"); /*check file created or not*/
 if(fp==NULL)
{ printf("File does not created!!!");
exit(0); /*exit from program*/ }
 printf("File created successfully."); /*writting into file*/
 putc('A',fp);
putc('B',fp);
putc('C',fp);
printf("\nData written successfully.");
 fclose(fp); /*again open file to read data*/
fp=fopen(fName,"r");
 if(fp==NULL)
{
 printf("\nCan't open file!!!");
 exit(0);
 }
printf("Contents of file is :\n");
 printf("%c",getc(fp));
printf("%c",getc(fp));
printf("%c",getc(fp));
 fclose(fp);
 return 0;
}
```

WEEK14: Objective: To understand data files and file handling with various file I/O functions. Explore the differences between text and binary files.

Suggested Experiments/Activities:

Tutorial 14: File handling

Lab 14: File operations

i) Write a C program to write and read text into a file.

```
#include< stdio.h >

int main()

{


   FILE *fp;   /* file pointer*/

   char fName[20];


   printf("\nEnter file name to create :");

   scanf("%s",fName);


   /*creating (open) a file*/

   fp=fopen(fName,"w");

   /*check file created or not*/

   if(fp==NULL)

   {

      printf("File does not created!!!");

      exit(0); /*exit from program*/

   }


   printf("File created successfully.");

   /*writting into file*/

   putc('A',fp);

   putc('B',fp);
```

```
    putc('C',fp);


    printf("\nData written successfully.");
    fclose(fp);


    /*again open file to read data*/
    fp=fopen(fName,"r");
    if(fp==NULL)
    {
        printf("\nCan't open file!!!");
        exit(0);
    }


    printf("Contents of file is :\n");
    printf("%c",getc(fp));
    printf("%c",getc(fp));
    printf("%c",getc(fp));


    fclose(fp);
    return 0;
}
```

Output

```
Enter file name to create : ok.txt
File created successfully.
Data written successfully.
Contents of file is :
ABC
```

ii) Write a C program to write and read text into a binary file using fread() and fwrite()

```
#include<stdio.h>
struct student{
   int sno;
   char sname [30];
   float marks;
   char temp;
};
main ( ){
   struct student s[60];
   int i;
   FILE *fp;
   fp = fopen ("student1.txt", "w");
   for (i=0; i<2; i++){
      printf ("enter details of student %d
", i+1);
      printf("student number:");
      scanf("%d",&s[i].sno);
      scanf("%c",&s[i].temp);
      printf("student name:");
      gets(s[i].sname);
      printf("student marks:");
      scanf("%f",&s[i].marks);
      fwrite(&s[i], sizeof(s[i]),1,fp);
   }
   fclose (fp);
   fp = fopen ("student1.txt", "r");
   for (i=0; i<2; i++){
```

```
    printf ("details of student %d are", i+1);
    fread (&s[i], sizeof (s[i]) ,1,fp);
    printf("student number = %d", s[i]. sno);
    printf("student name = %s", s[i]. sname);
    printf("marks = %f", s[i]. marks);
  }
  fclose(fp);
  getch( );
}
```

Output

```
enter details of student 1
student number:1
student name:pinky
student marks:56
enter details of student 2
student number:2
student name:rosy
student marks:87
details of student 1 are
student number = 1
student name = pinky
marks = 56.000000
details of student 2 are
student number = 2
student name = rosy
marks = 87.000000
```

iii) Copy the contents of one file to another file.

```
#include <stdio.h>
#include <stdlib.h> // For exit()

int main()
{
```

```
FILE *fptr1, *fptr2;
char filename[100], c;

printf("Enter the filename to open for reading \n");
scanf("%s", filename);

// Open one file for reading
fptr1 = fopen(filename, "r");
if (fptr1 == NULL)
{
   printf("Cannot open file %s \n", filename);
   exit(0);
}

printf("Enter the filename to open for writing \n");
scanf("%s", filename);

// Open another file for writing
fptr2 = fopen(filename, "w");
if (fptr2 == NULL)
{
   printf("Cannot open file %s \n", filename);
   exit(0);
}

// Read contents from file
c = fgetc(fptr1);
while (c != EOF)
```

```
    {
        fputc(c, fptr2);
        c = fgetc(fptr1);
    }

    printf("\nContents copied to %s", filename);

    fclose(fptr1);
    fclose(fptr2);
    return 0;
}
```

Output:

```
Enter the filename to open for reading
a.txt
Enter the filename to open for writing
b.txt
Contents copied to b.txt
```

iv) Write a C program to merge two files into the third file using command-line arguments.

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

int main()
{
  // Open two files to be merged
  FILE *fp1 = fopen("file1.txt", "r");
  FILE *fp2 = fopen("file2.txt", "r");
```

```
  // Open file to store the result
  FILE *fp3 = fopen("file3.txt", "w");
  char c;

  if (fp1 == NULL || fp2 == NULL || fp3 == NULL)
  {
      puts("Could not open files");
      exit(0);
  }

  // Copy contents of first file to file3.txt
  while ((c = fgetc(fp1)) != EOF)
    fputc(c, fp3);

  // Copy contents of second file to file3.txt
  while ((c = fgetc(fp2)) != EOF)
    fputc(c, fp3);

  printf("Merged file1.txt and file2.txt into file3.txt");

  fclose(fp1);
  fclose(fp2);
  fclose(fp3);
  return 0;
}
```

Output:

Merged file1.txt and file2.txt into file3.txt

v) Find no. of lines, words and characters in a file

```
#include <stdio.h>
int main()
{
   char str[100];//input string with size 100

   int words=0,newline=0,characters=0; // counter variables

   scanf("%[^~]",&str);//scanf formatting

   for(int i=0;str[i]!='\0';i++)
    {
       if(str[i] == ' ')
       {
           words++;
       }
       else if(str[i] == '\n')
       {
          newline++;
           words++;//since with every next line new words start. corner case 1
       }
       else if(str[i] != ' ' && str[i] != '\n'){
       characters++;
       }
    }
   if(characters > 0)//Corner case 2,3.
   {
      words++;
```

```
        newline++;
    }
     printf("Total number of words : %d\n",words);
     printf("Total number of lines : %d\n",newline);
     printf("Total number of characters : %d\n",characters);
    return 0;
}
```

Input : Deepak Kumar is the
brother of Aditya Kumar~
Output : Total number of words : 8
Total number of lines : 2
Total number of characters : 36

vi) Write a C program to print last n characters of a given file.

```
#include<stdio.h>
void main()
{
FILE *fp;
char ch;
int n;
long len;
clrscr();
printf("Enter the value of n : ");
scanf("%d",&n);
fp=fopen("test.txt","r");
if(fp==NULL)
    {
    puts("cannot open this file");
    exit(1);
```

```
    }

fseek(fp,0,SEEK_END);

len = ftell(fp);

fseek(fp,(len-n),SEEK_SET);

do {
  ch = fgetc(fp);
  putchar(ch);
}while(ch!=EOF);

fclose(fp);
getch();
}
```

Output :

```
Enter the value of n : 4
.com
```