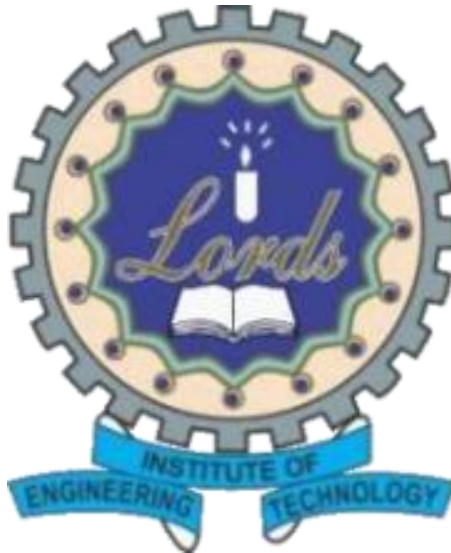


LORDS

Institute of Engineering & Technology (A UGC Autonomous Institution)

Approved by AICTE / Affiliated to Osmania University / Estd.2002.

Accredited 'A' grade by NAAC, Accredited by NBA



B.E I/II SEMESTER

PROGRAMMING FOR PROBLEM SOLVING LAB MANUAL (Common to all branches)

Name:_____

Roll No.:_____

Branch & Section:_____



LORDS INSTITUTE OF ENGINEERING & TECHNOLOGY

(UGC Autonomous)


Approved by AICTE | Affiliated to Osmania University | Estd.2003.

Vision of the Institute:

Lords Institute of Engineering and Technology strives for excellence in professional education through quality, innovation and teamwork and aims to emerge as a premier institute in the state and across the nation.

Mission of the Institute:

- To impart quality professional education that meets the needs of present and emerging technological world.
- To strive for student achievement and success, preparing them for life, career and leadership.
- To provide a scholarly and vibrant learning environment that enables faculty, staff and students to achieve personal and professional growth.
- To contribute to advancement of knowledge, in both fundamental and applied areas of engineering and technology.
- To forge mutually beneficial relationships with government organizations, industries, society and the alumni.


Principal
PRINCIPAL
Lords Institute of Engineering & Tech
Sy.No. 32, Himayath Sagar, Hyderabad-500 081



LORDS INSTITUTE OF ENGINEERING & TECHNOLOGY

(UGC Autonomous)

Approved by AICTE | Affiliated to Osmania University | Estd.2003.

Department of Computer Science and Engineering

Vision of the Department:

To emerge as a centre of excellence by imparting quality technical education through innovation, team work & value creation, and to contribute to advancement of knowledge in the field of Computer Science & Engineering.

Mission of the Department:

DM1: Providing the students with in-depth understanding of fundamentals and practical training related to professional skills, problem solving skills and their applications through effective Teaching-Learning Process and State of the art laboratories pertaining to computer science & engineering and inter-disciplinary areas.

DM2: Preparing students in developing research, design, entrepreneurial skills and employability capabilities.

DM3: Providing consultancy services and promoting Industry-Department Interactions.

Note: DM: Department Mission

Shankar J. J.
Head of the Department

Head of the Department
Computer Science and Engineering
Lords Institute of Engg & Tech
Hyderabad - 500091, T.S.



LORDS INSTITUTE OF ENGINEERING & TECHNOLOGY


(UGC Autonomous)

Approved by AICTE | Affiliated to Osmania University | Estd.2003.

Department of Computer Science and Engineering

B.E. Computer Science and Engineering Program Educational Objectives (PEOs):

PEO1	Shall have strong foundations in Basic Sciences, Mathematics, Computer Science and allied engineering.
PEO2	Shall be capable of identifying, formulating, analyzing and creating and communicating engineering solutions using appropriate modern engineering techniques, designing skills and tools to develop novel products solutions and simulations for the real life problems in Computer Science and Engineering.
PEO3	Shall have successful and productive engineering careers, with emphasis on technical competency and managerial skills so that they are really accepted by the industry with minimal orientations.
PEO4	Shall have professional, project management skills, ethics, research skills and leadership for independent working or team spirit to work cohesively within a group.


Head of the Department
Head of the Department
Computer Science and Engineering
Lords Institute of Engg & Tech
Hyderabad - 500091, T.S.



LORDS INSTITUTE OF ENGINEERING & TECHNOLOGY

(UGC Autonomous)

Approved by AICTE | Affiliated to Osmania University | Estd.2003.

Department of Computer Science and Engineering

B.E. Computer Science and Engineering Program Outcomes (POs):

Engineering Graduates will be able to:

S.No.	Program Outcomes (POs):
1.	Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2.	Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3.	Design/Development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4.	Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5.	Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6.	The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7.	Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8.	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9.	Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10.	Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11.	Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12.	Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Srinivasa Reddy
Head of the Department

Head of the Department
Computer Science and Engineering
Lords Institute of Engg & Tech
Hyderabad - 500091, T.S.



LORDS INSTITUTE OF ENGINEERING & TECHNOLOGY
(UGC Autonomous)

Approved by AICTE | Affiliated to Osmania University | Estd.2003.

Department of Computer Science and Engineering

B.E. Computer Science and Engineering Program Specific Outcomes (PSO's):

PSO1	Professional Skills: The ability to research, understand and implement computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics and networking for efficient analysis and design of computer-based systems of varying complexity
PSO2	Problem-Solving Skills: The ability to apply standard practices and strategies in software service management using open-ended programming environment with agility to deliver a quality service for business success

slau gupel
Head of the Department
Head of the Department
Computer Science and Engineering
Lords Institute of Engg & Tech
Hyderabad - 500091, T.S.



LORDS INSTITUTE OF ENGINEERING & TECHNOLOGY [A]

Approved by AICTE / Affiliated to Osmania University / Estd.2003.

Accredited 'A' grade by NAAC, Accredited by NBA

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

LABORATORY MANUAL PROGRAMMING FOR PROBLEM SOLVING

Prepared by
Mr. Tammisetty Vijay Kumar
Assistant Professor(CSE)
Mrs. Pooja Chavan
Assistant Professor(CSE)

LORDS INSTITUTE OF ENGINEERING & TECHNOLOGY[A]

B.E I/II SEMESTER
(Common to ALL Branches)

PROGRAMMING FOR PROBLEM SOLVING LABORATORY

Course Code	Hours / Week	Credits-2			Maximum Marks		
U21CS1L1	L	T	D	P	CIA	SEE	Total
	-	-	-	4	25	50	75
	Practical Classes: 36				Total Classes: 36		

Course Objectives:

The course should enable the students to:

1. To understand the fundamentals of programming in C language.
2. To write, compile and debug programs in C.
3. To formulate solutions to problems and implement in C .
4. To effectively choose programming components to solve computing problems.

Course Outcomes:

On completion of this Course, Students are able to:

1. Choose appropriate data type for implementing programs in C language.
2. Design and implement modular programs involving input output operations, decision making and looping constructs.
3. Implement search and sort operations on arrays.
4. To decompose a problem into functions and to develop modular reusable code.
5. Apply the concept of pointers for implementing programs on dynamic memory management and string handling.
6. Design and implement programs to store data in structures and files.

LIST OF EXPERIMENTS

1	Basic Programs	Page 1-3
<ol style="list-style-type: none">a. Write a C program to generate all prime numbers between 1 and n.b. Write a C program to check whether the given number is Armstrong or not.c. Write a C program to calculate all arithmetic operations using switch case.		
2	MAX and MIN of set of given numbers & finding roots of quadratic equation	Page 4-6
<ol style="list-style-type: none">a. Write a C program to find roots of a Quadratic Equation.b. Write a C program to find maximum and minimum of given list of numbers.		
3	Sin x & Cos x values using series expansion	Page 7-8
<ol style="list-style-type: none">a. Write a C program to compute sine x for given x.b. Write a C program to compute cosine of x for given x.		

4	Conversion of Binary to Decimal, Octal, Hexadecimal and vice versa	Page 9-13
a. Write a C program to convert Binary to Decimal. b. Write a C program to convert Decimal to Binary. c. Write a C program to convert Decimal to Hexadecimal. d. Write a C program to convert Hexadecimal to Octal.		
5	Generating Pascal Triangle & Pyramid of Numbers	Page 14-16
a. Write a C program to generate the following pattern. <pre> 1 1 2 1 2 3 1 2 3 4 </pre> b. Write a C program to generate the following pattern. <pre> * * * * * * * * * * </pre> c. Write a C program to generate Pascal Triangle up to N rows.		
6	Programs on Recursions	Page 17-19
Q) Write a C program that uses recursion to find: i) GCD ii) Factorial iii) Fibonacci		
7	I. Matrix Addition & Multiplication using Arrays II. Linear and Binary Search	Page 20-26
a. Write a C program to perform addition of two matrices. b. Write a C program to perform multiplication of two matrices. c. Write a C program to find an element in an Array using Linear Search. d. Write a C program to find an element in an Array using Binary Search.		
8	Bubble Sort & Selection Sort	Page 27-28
a. Write a C program to sort the elements in an Array using Bubble sort in ascending order. b. Write a C program to sort the elements in an Array using Selection sort in ascending order.		

9	Programs on Pointers: Pointers to Arrays, Pointer to Function	Page 29-32
a. Write a C program to concatenate two strings using pointers. b. Write a C program to find the length of string using pointers. c. Write a C program to compare two strings using pointers. d. Write a C program to swap two numbers using call by reference.		
10	Programs on String Manipulation	Page 33-35
a. Write a C Program to calculate length of a string <ul style="list-style-type: none"> i. without using Built in String Function. ii. with using String Functions b. Write a C Program to display reverse of a string <ul style="list-style-type: none"> i. without using Built in String Function. ii. with using String Functions c. Write a C program to determine if the given string is a palindrome or not.		
11	Programs on Structures and Unions	Page 36-39
a. Write a C program to accept and display student details(Rollno, Name, Branch,Address) using structures. b. Create a union containing 6 strings: name, home_ address, hostel_ address, city, state and zip. Write a C program to display your present address.		
12	File Handling Programs	Page 40-43
a. Write a C program to display the contents of a file. b. Write a C program to copy the contents of one file to another. c. Write a C program to count no of lines, words and characters in a given text file.		
Reference Books:		
<ol style="list-style-type: none"> 1. Byron Gottfried, "Theory and practice of Programming with C", Schaum's Outline, McGraw-Hill 1996. 2. A.K. Sharma, -"Computer Fundamentals and Programming in C", Universities Press, 2nd Edition, 2018. 3. E. Balaguruswamy –"Programming in ANSI C", Tata McGraw-Hill Education, 2008. 4. Brian W. Kernighan and Dennis M. Ritchie, "The C Programming Language", Prentice Hall of India 1988. 		
Web References:		
<ol style="list-style-type: none"> 1. http://www.geeksforgeeks.org/c 2. http://www.cprogramming.com/tutorial/c 3. http://www.cs.princeton.edu 		

1.BASIC PROGRAMS

- a. Write a program to check whether the given number is Armstrong or not

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,r,sum=0,temp;
    //clrscr();
    printf("\n Enter the number");
    scanf("%d",&n);
    temp=n;
    while(n>0)
    {
        r=n%10;
        sum=sum+(r*r*r);
        n=n/10;
    }
    if(temp==sum)
        printf("\n Armstrong Number\n");
    else
        printf(" \n Not Armstrong Number\n");
    getch();
}
```

Output:

Enter the number

153

Armstrong Number

Enter the Number

131

Not Armstrong Number

b. Write a program to generate all prime numbers between 1 and n.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,i,j,count;
    clrscr();
    printf("\n prime number of series \n");
    printf("\n Enter any number \n");
    scanf("%d",&n);
    printf("\n The prime number is between 1 to %d\n",n);
    for(i=1;i<=n;i++)
    {
        count=0;
        for(j=1;j<=i;j++)
            if(i%j==0)
            {
                count++;
            }
        if(count==2)
        {
            printf("%d\t",i);
        }
    }
    getch();
}
```

Output:

Prime number of series

Enter any number

30

The prime number is between 1 to 30

2 3 5 7 11 13 17 19 23 29

c. Write a C program to calculate all arithmetic operations using switch case.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a,b,c; char ch;
    clrscr();
    printf("Enter operator(either +,-,*,/)\n");
    scanf("%c",&ch);
    printf("Enter Two numbers\n");
    scanf("%d%d",&a,&b);
    switch(ch)
    {
        case '+':    c=a+b;
                    printf("Addition Result=%d\n",c);
                    break;
        case '-':    c=a-b;
                    printf("Subtraction Result=%d\n",c);
                    break;
        case '*':    c=a*b;
                    printf("Multiplication Result=%d\n",c);
                    break;
        case '/':    c=a/b;
                    printf("Division Result=%d\n",c);
                    break;
        case '%':    c=a%b;
                    printf("Modulo Division Result=%d\n",c);
                    break;
        default: printf("Invalid Operator\n");
    }
    getch();
}
```

Output:

Enter operator (either +,-,*,/)

*

Enter two numbers

3 5

Multiplication Result = 15

2.MAX and MIN of set of given numbers & finding roots of quadratic equation

- a. Write a C program to find roots of a Quadratic Equation.

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

main()
{
    float a,b,c,dis,r1,r2;//a,b,c are constant coefficients, dis= discriminant and r1,r2 are roots
    printf("\nQuadratic Equation is of the form: ax^2 + bx + c = 0 \n");
    printf("\nEnter the values of a, b and c: ");
    scanf("%f %f %f",&a,&b,&c);

    dis = pow(b,2) - 4*a*c; // calculation of discriminant
    if(dis < 0) //checking the value of discriminant
    {
        printf("\nThe roots are imaginary.\n\n");
        printf("Root1= %.3f % + .3fi",-b/(2*a),sqrt(-dis)/(2*a));
        printf("\nRoot2= %.3f % + .3fi\n",-b/(2*a),-sqrt(-dis)/(2*a));
    }
    else
    {
        r1 = (-b + sqrt(dis))/(2.0*a);
        r2 = (-b - sqrt(dis))/(2.0*a);
        printf("\nThe first root is = %f\nThe second root is = %f\n",r1,r2);
    }
    return 0;
}
```


Output:

```
Quadratic Equation is of the form:  $ax^2 + bx + c = 0$   
Enter the values of a, b and c: 1 -5 5  
The first root is = 3.618034  
The second root is = 1.381966
```

```
Quadratic Equation is of the form:  $ax^2 + bx + c = 0$   
Enter the values of a, b and c: 4 5 6  
The roots are imaginary.  
Root1= -0.625 +1.053i  
Root2= -0.625 -1.053i
```

b. Write a program to find maximum and minimum of given list of numbers.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int arr[100],max,min,i,N;
    clrscr();
    printf("\nEnter the size of N elements \n");
    scanf("%d",&N);
    printf("\nEnter %d values ",N);
    for(i=0;i<N;i++)
    {
        scanf("%d",&arr[i]);
    }
    max=min=arr[0];
    for(i=0;i<N;i++)
    {
        if(arr[i]>max)
            max=arr[i];
        if(arr[i]<min)
            min=arr[i];
    }
    printf("\nBiggest number among %d elements=%d",N,max);
    printf("\n Smallest number among %d elements=%d\n",N,min);
    getch();
}
```

Output:

Enter the size of N elements 5

Enter 5 values 50 -4 43 -32 34

Biggest number among 5 elements= 50

Smallest number among 5 elements= -32

3.Sin x & Cos x values using series expansion

a. Write a C program to compute sine x for given x.

The user should supply x and a positive integer n. We compute the sine of x using the series and the computation should use all terms in the series up through the term involving x^n

$$\sin x = x - x^3/3! + x^5/5! - x^7/7! + x^9/9! \dots\dots$$

```
#include <stdio.h>
#include<conio.h>
void main()
{
    int i, j, n, fact, sign = - 1;
    float x, p, sum = 0;
    clrscr();
    printf("\n\nEnter the value of x : ");
    scanf("%f", &x);
    printf("Enter the value of n : ");
    scanf("%d", &n);
    for (i = 1; i <= n; i += 2)
    {
        p = 1;
        fact = 1;
        for (j = 1; j <= i; j++)
        {
            p = p * x;
            fact = fact * j;
        }
        sign = - 1 * sign;
        sum += sign * p / fact;
    }
    printf("The value of Sin %0.2f = %f", x, sum);
    getch();
}
```

Output:

```
Enter the value of x : 2
Enter the value of n : 3
The value of Sin 2.00 = 0.666667
```


b. Write a program to compute cosine of x for given x.

The user should supply x and a positive integer n. We compute the cosine of x using the series and the computation should use all terms in the series up through the term involving x^n

$$\cos x = 1 - x^2/2! + x^4/4! - x^6/6! \dots$$

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i, j, n, fact, sign = - 1;
    float x, p, sum = 0;
    clrscr();
    printf("Enter the value of x : ");
    scanf("%f", &x);
    printf("Enter the value of n : ");
    scanf("%d", &n);
    for (i = 2; i <= n; i += 2)
    {
        p = 1;
        fact = 1;
        for (j = 1; j <= i; j++)
        {
            p = p * x;
            fact = fact * j;
        }
        sum += sign * p / fact;
        sign = - 1 * sign;
    }

    printf("The value of Cos %0.2f = %f", x, 1+sum);
    getch();
}
```

Output:

Enter the value of x : 4

Enter the value of n : 3

The value of Cos 4.00 = -7.000000

4.Conversion of Binary to Decimal,Octal,Hexadecimal and vice versa

a. Write a C program to convert Binary to Decimal.

```
#include <stdio.h>
#include <math.h>
int binaryToDecimal(long binarynum)
{
    int decimalnum = 0, temp = 0, remainder;
    while (binarynum!=0)
    {
        remainder = binarynum % 10;
        binarynum = binarynum / 10;
        decimalnum = decimalnum + remainder*pow(2,temp);
        temp++;
    }
    return decimalnum;
}
int main()
{
    long binarynum;
    //clrscr();
    printf("\n\nEnter a binary number: ");
    scanf("%ld", &binarynum);
    printf("Equivalent decimal number is: %d", binaryToDecimal(binarynum));
    getch();
    return 0;
}
```

Output:

Enter a binary number: 1010111

Equivalent decimal number is: 87

b. Write a C program to convert Decimal to Binary.

```
// convert decimal to binary

#include <stdio.h>
#include <math.h>
long long convert(long long);
void main()
{
    long long n, bin;
    clrscr();
    printf("\n\nEnter a decimal number: ");
    scanf("%lld", &n);
    bin = convert(n);
    printf("%lld in decimal = %lld in binary", n, bin);
    getch();
}

long long convert(long long n)
{
    long long bin = 0;
    long long rem, i = 1;
    while (n!=0)
    {
        rem = n % 2;
        n /= 2;
        bin =bin+ (rem * i);
        i =i*10;
    }
    return bin;
}
```

Output:

Enter a Decimal Number: 26

26 in decimal = 11010 in binary

c. Write a C program to convert Decimal to Hexadecimal.

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int decnum, rem, i=0;
    char hexnum[50];
    // clrscr();
    printf("\n\nEnter any decimal number: ");
    scanf("%d", &decnum);
    while(decnum!=0)
    {
        rem = decnum%16;
        if(rem<10)
            rem = rem+48;
        else
            rem = rem+55;
        hexnum[i] = rem;
        i++;
        decnum = decnum/16;
    }
    printf("\nEquivalent Value in Hexadecimal = ");
    for(i=i-1; i>=0; i--)
        printf("%c", hexnum[i]);
    getch();
    return 0;
}
```

Output:

Enter any Decimal Number: 172

Equivalent value in Hexadecimal= AC

Enter any Decimal Number: 64189

Equivalent value in Hexadecimal= FABD

d. Write a C program to convert Hexadecimal to Octal.

```
// C Program for Hexadecimal to Octal Conversion

#include<stdio.h>
#include<string.h>
#include<conio.h>
#include<math.h>
int convert(char hexa[])
{
    int i,size, deci=0, octa=0;
    for(size=0; hexa[size]!='\0'; size++); //this loop calculates size of hexadecimal number
    for(i=0; hexa[i]!='\0'; i++,size--)
    {
        if(hexa[i]>='0' && hexa[i]<='9')
        {
            deci= deci + (hexa[i]-'0')*pow(16,size-1);
        }
        if(hexa[i]>='A' && hexa[i]<='F')
        {
            deci = deci + (hexa[i]-55)*pow(16,size-1);
        }
        if(hexa[i]>='a' && hexa[i]<='f')
        {
            deci = deci + (hexa[i]-87)*pow(16,size-1);
        }
    } // deci contains the decimal value of given hexadecimal number.
    i=1;
    while(deci!=0)
    {
        octa = octa + (deci%8)*i;
        deci = deci/8;
        i = i*10;
    }
    return octa;
}
```

```
int main()
{
    char hexa[20];
    clrscr();
    printf("Enter Hexadecimal Number : ");
    scanf("%s",hexa);
    printf("Equivalent Octal Value = %d",convert(hexa));
    getch();
    return 0;
}
```

Output:

Enter any Hexadecimal Number: 9
Equivalent Octal value = 11

Enter any Hexadecimal Number: F
Equivalent Octal value = 17

Enter any Hexadecimal Number: A
Equivalent Octal value = 12

Enter any Hexadecimal Number: 10
Equivalent Octal value = 20

5. Generating Pascal Triangle & Pyramid of Numbers

- a. Write a program to generate the following pattern.

```
1
1 2
1 2 3
1 2 3 4
```

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i,j,N;
    clrscr();
    printf("Enter size of N");
    scanf("%d",&N);

    printf("Pattern upto %d rows is\n",N);
    for(i=1;i<=N;i++)
    {
        for(j=1;j<=i;j++)
        {
            printf("%d ",j);
        }
        printf("\n");
    }
    getch();
}
```

Output:

Enter size of N 5

Pattern upto 5 rows is

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

b. Write a program to generate the following pattern.

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

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i,j,N;
    clrscr();
    printf("Enter size of N ");
    scanf("%d",&N);
    printf("Pattern upto %d rows is\n",N);
    for(i=1;i<=N;i++)
    {
        for(j=1;j<=i;j++)
        {
            printf("* ");
        }
        printf("\n");
    }
    getch();
}
```

Output:

Enter size of N 5

Pattern upto 5 rows is

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


c. Write a program to generate Pascal Triangle up to N rows.

```
#include <stdio.h>
long fun(int y)
{
    int z;
    long result = 1;

    for( z = 1 ; z <= y ; z++ )
        result = result*z;

    return ( result );
}
int main()
{
    int x, y, z;
    printf("Input the number of rows in Pascal's triangle: ");
    scanf("%d",&y);
    for ( x = 0 ; x < y ; x++ )
    {
        for ( z = 0 ; z <= ( y - x - 2 ) ; z++ )
            printf(" ");
        for( z = 0 ; z <= x ; z++ )
            printf("%ld ",fun(x)/(fun(z)*fun(x-z)));

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

Output:

```
Input the number of rows in Pascal's triangle: 5
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
```

6.Programs on Recursions

- a. Write a program that uses recursion to find: a) GCD b) Factorial c) Fibonacci

a)GCD Program:

```
#include<stdio.h>
#include<conio.h>
int gcd(int n1,int n2);
void main()
{
    int a,b,result;
    clrscr();
    printf("enter a and b values\n");
    scanf("%d%d",&a,&b);
    result=gcd(a,b);
    printf("GCD of %d and %d is %d\n",a,b,result);
    getch();
}
int gcd(int n1,int n2)
{
    if(n2!=0)
        return gcd(n2,n1%n2);
    return n1;
}
```

Output:

```
enter a and b values
40  20
GCD of 40 and 20 is 20
```

b)Factorial Program:

```
#include<stdio.h>

#include<conio.h>

long double fact(long double k);

void main()
{
    long double N,result;

    clrscr();

    printf("\n\nEnter N");

    scanf("%Lf",&N);

    result=fact(N);

    printf("Factorial of %Lf is %Lf\n",N,result);

    getch();
}

long double fact(long double k)
{
    if(k==0||k==1)
        return 1;

    return k*fact(k-1);
}
```

Output:

Enter N 5

Factorial of 5 is 120

c)Fibonacci Series Program:

```
#include<stdio.h>
#include<conio.h>
int fib(int k);
void main()
{
    int N,i;
    clrscr();
    printf("Enter Nth term for Fibonacci Series: ");
    scanf("%d",&N);
    printf("Fibonacci series up to %d is\n",N);
    for(i=0;i<N;i++)
    {
        printf("%d ",fib(i));
    }
    printf("\n");
    getch();
}
int fib(int k)
{
    if(k==0)
        return 0;
    if(k==1)
        return 1;
    return fib(k-1)+fib(k-2);
}
```

Output:

Enter Nth term for Fibonacci Series: 5

Fibonacci series up to 5 is

0 1 1 2 3

7. Matrix Addition & Multiplication using Arrays

a. Write a program to perform addition of two matrices.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[100][100],b[100][100],c[100][100],i,j;
    int rows,cols;
    clrscr();
    printf("Enter rows and columns for matrix\n");
    scanf("%d%d",&rows,&cols);
    printf("Enter values for 1st matrix\n");
    for(i=0;i<rows;i++)
    {
        for(j=0;j<cols;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
    printf("\nEnter values for 2nd matrix\n");
    for(i=0;i<rows;i++)
    {
        for(j=0;j<cols;j++)
        {
            scanf("%d",&b[i][j]);
        }
    }

    for(i=0;i<rows;i++)
    {
        for(j=0;j<cols;j++)
        {
            c[i][j]=a[i][j]+b[i][j];
        }
    }
    printf("Matrix Addition is\n");
    for(i=0;i<rows;i++)
    {
        for(j=0;j<cols;j++)
        {
            printf("\t%d ",c[i][j]);
        }
        printf("\n");
    }
    getch();
}
```

Output:

Enter rows and columns for matrix

3 3

Enter values for 1st matrix

3 3 3

3 3 3

3 3 3

Enter values for 2nd matrix

3 3 3

3 3 3

3 3 3

Matrix Addition is

6 6 6

6 6 6

6 6 6

b. Write a program to perform multiplication of two matrices.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[100][100],b[100][100],c[100][100],i,j,k;
    int r1,r2,c1,c2;
    clrscr();
    printf("Enter rows and columns of 1st matrix\n");
    scanf("%d%d",&r1,&c1);
    printf("Enter values for 1st matrix\n");
    for(i=0;i<r1;i++)
    {
        for(j=0;j<c1;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
    printf("Enter rows and columns of 2nd matrix\n");
    scanf("%d%d",&r2,&c2);
    if(c1!=r2)
        printf("Matrix multiplication not possible\n");
    else
    {
        printf("Enter values for 2nd matrix\n");
        for(i=0;i<r2;i++)
        {
            for(j=0;j<c2;j++)
            {
                scanf("%d",&b[i][j]);
            }
        }
        for(i=0;i<r1;i++)
        {
            for(j=0;j<c1;j++)
            {
                c[i][j]=0;
                for(k=0;k<c2;k++)
                {
                    c[i][j]+=a[i][k]*b[k][j];
                }
            }
        }
    }
}
```

```
        printf("\nMatrix product is\n");
        for(i=0;i<r1;i++)
        {
            for(j=0;j<c2;j++)
            {
                printf("%d ",c[i][j]);
            }
            printf("\n");
        }
    }
    getch();
}
```

Output:

Enter rows and columns of 1st matrix

2 2

Enter values for 1st matrix

3 3

3 3

Enter rows and columns of 2nd matrix

2 2

Enter values for 1st matrix

4 4

4 4

Matrix product is

24 24

24 24

Linear and Binary Search

c. Write a program to find an element in an Array using Linear Search.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int arr[100],N,i,key,flag=1;
    clrscr();
    printf("Enter the size of N elements ");
    scanf("%d",&N);
    printf("Enter %d values into array ",N);
    for(i=0;i<N;i++)
        scanf("%d",&arr[i]); printf("\nArray is\n");
    for(i=0;i<N;i++)
        printf("%d ",arr[i]);
    printf("\nEnter key value to find ");
    scanf("%d",&key);
    for(i=0;i<N;i++)
    {
        if(key==arr[i])
        {
            printf("Value found at %d position\n",i+1);
            flag=0; break;
        }
    }

    if(flag)
        printf("Value not found\n");
    getch();
}
```

Output

```
Enter the size of N elements 5
Enter 5 values into array
10    20    40    30    50
Enter key value to find
40
Value found at 3 position
```

d. Write a program to search an element in an Array using Binary Search.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int N,arr[100],key,i,flag=1,low,high,mid;
    clrscr();
    printf("Enter N ");
    scanf("%d",&N);
    printf("\nEnter the size of N elements ");
    for(i=0;i<N;i++)
        scanf("%d",&arr[i]);
    printf("\nArray is\n");
    for(i=0;i<N;i++)
        printf("%d ",arr[i]);
    printf("\nEnter key to find\n");
    scanf("%d",&key);
    low=0;
    high=N-1;
    mid=(low+high)/2;
    for(;low<=high;mid=(low+high)/2)
    {
        if(arr[mid]==key)
        {
            printf("Value found at %d position\n",mid+1);
            flag=0;
            break;
        }

        if(arr[mid]>key)
            high=mid-1;
        if(arr[mid]<key)
            low=mid+1;
    }
    if(flag)
        printf("Value Not found\n");
    getch();
}
```

Output:

Enter the size of N elements 8

Enter 8 values

10 20 30 40 50 60 70 80

Array is

10 20 30 40 50 60 70 80

Enter key to find

70

Value found at 7 position

8. Bubble Sort & Selection Sort

a. Write a program to sort the elements in an Array using Bubble sort in ascending order.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int N,arr[100],i,j,temp;
    clrscr();
    printf("Enter N ");
    scanf("%d",&N);
    printf("Enter %d values \n",N);
    for(i=0;i<N;i++)
        scanf("%d",&arr[i]);
    printf("\nArray before sort is\n");
    for(i=0;i<N;i++)
        printf("%d ",arr[i]);
    for(i=0;i<N;i++)
    {
        for(j=0;j<N-i-1;j++)
        {
            if(arr[j]>arr[j+1])
            {
                temp=arr[j];
                arr[j]=arr[j+1];
                arr[j+1]=temp;
            }
        }
    }
    printf("\n Array after using bubble sort is \n");
    for(i=0;i<N;i++)
        printf("%d ",arr[i]);
    getch();
}
```

Output:

```
Enter the value of n  5
Enter 5 values
40    30    20    10    5
Array before sort is
40    30    20    10    5
Array after using bubble sort is
5     10    20    30    40
```


b. Write a program to sort the elements in an Array using Selection sort.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int N,arr[100],i,j,temp,min;
    clrscr();
    printf("\nEnter the size of N elements to store ");
    scanf("%d",&N);
    printf("Enter %d values \n",N);
    for(i=0;i<N;i++)
        scanf("%d",&arr[i]);
    printf("\nArray before sort is\n");
    for(i=0;i<N;i++)
        printf("%d ",arr[i]);
    for(i=0;i<N;i++)
    {
        min=i;
        for(j=i+1;j<N;j++)
        {
            if(arr[j]<arr[min])
            {
                min=j;
            }
        }
        temp=arr[i];
        arr[i]=arr[min];
        arr[min]=temp;
    }
    printf("\n Array after using Selection sort is\n");
    for(i=0;i<N;i++)
        printf("\t%d ",arr[i]);
    getch();
}
```

Output:

```
Enter N 5
Enter 5 values
40 30 20 10 5
Array before sort is
40 30 20 10 5
Array after using Selection sort is
5 10 20 30 40
```

9. Programs on Pointers: Pointers to Arrays, Pointer to Function

a. Write a C program to concatenate two strings using pointers.

```
#include <stdio.h>

#define MAX_SIZE 100 // Maximum string size

int main()
{
    char str1[MAX_SIZE], str2[MAX_SIZE];
    char * s1 = str1;
    char * s2 = str2;
    // Inputting 2 strings from user
    //clrscr();
    printf("\nEnter 1st string: ");
    gets(str1);
    printf("\nEnter 2nd string: ");
    gets(str2);
    // Moving till the end of str1
    while(*(++s1));
    // Coping str2 to str1
    while(*(s1++) = *(s2++));
    printf("\nConcatenated string: %s", str1);
    getch();
    return 0;
}
```

Output:

Enter 1st string: LORDS

Enter 1st string: COLLEGE

Concatenated string: LORDSCOLLEGE

b. Write a C program to find the length of string using pointers.

```
#include<stdio.h>
#include<conio.h>
int main()
{
    char str[100], *pt;
    int i = 0;
    clrscr();
    printf("\nPointer Example Program : Find or Calculate Length of String \n");
    printf("\nEnter Any string [below 100 chars] : ");
    gets(str);
    pt = str;
    while (*pt != '\0')
    {
        i++;
        pt++;
    }
    printf("\nLength of String : %d", i);
    getch();
    return 0;
}
```

Output:

```
Pointer Example Program : Find or Calculate Length of String
Enter Any string [below 100 chars] : LORDS-COLLEGE
Length of String : 13
```

c. Write a C program to compare two strings using pointers.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    char string1[150],string2[150],*str1,*str2;
    int i;
    clrscr();
    printf("\nEnter The First String: ");
    scanf("%s",string1);
    printf("\nEnter The Second String: ");
    scanf("%s",string2);
    str1 = string1;
    str2 = string2;
    while(*str1 == *str2)
    {
        if ( *str1 == '\0' || *str2 == '\0' )
            break;
        str1++;
        str2++;
    }
    if( *str1 == '\0' && *str2 == '\0' )
        printf("\n\nBoth Strings Are Equal.");
    else
        printf("\n\nBoth Strings Are Not Equal.");
    getch();
}
```

Output:

Enter The First String: LORDS

Enter The Second String: LORDS

Both Strings Are Equal.

d. Write a program to swap two numbers using call by reference.

```
#include<stdio.h>
#include<conio.h>
void swap(int *x,int *y);
void main()
{
    int a,b;
    clrscr();
    printf("Enter a and b values\n");
    scanf("%d%d",&a,&b);
    printf("Before swap,a=%d and b=%d\n",a,b);
    swap(&a,&b);
    printf("After swap,a=%d and b=%d\n",a,b);
    getch();
}
void swap(int *x,int *y)
{
    int temp;
    temp=*x;
    *x=*y;
    *y=temp;
}
```

Output:

Enter a and b values

20 30

Before swap, a=20 and b=30

After swap, a=30 and b=20

10.Programs on String Manipulation

a. Write a C Program to calculate length of a string

i. without using Built in String Function.

ii. with using String Functions

(i)

```
#include <stdio.h>
#include<string.h>
int main()
{
    char str[100],i;clrscr();
    printf("\nEnter a string below [100] with characters: \n");
    gets(str);
    // '\0' represents end of String
    for(i=0; str[i]!='\0'; ++i);
    printf("\nLength of input string: %d",i);
    getch();
    return 0;
}
```

Output:

Enter a string below [100] with characters: VijayKumar

Length of input string: 10

(ii)

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
int main()
{
    char a[100];
    int length;
    clrscr();
    printf("\n\nEnter a string below [100] to calculate its length:\n");
    gets(a);
    length = strlen(a);
    printf("Length of the string = %d\n", length);
    getch();
    return 0;
}
```

Output:

Enter a string below [100] to calculate its length: Vijay Kumar Lords

Length of input string: 17

b. Write a C Program to display reverse of a string

i. without using Built in String Function.

ii. with using String Functions

(i)

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
    char string[140],temp;
    int i,length;
    clrscr();
    printf("\n\nEnter any String below [140] characters : ");
    scanf("%s",string);
    length=strlen(string)-1;
    for(i=0;i<strlen(string)/2;i++)
    {
        temp=string[i];
        string[i]=string[length];
        string[length--]=temp;
    }
    printf("\nReverse string :%s",string);
    getch();
}
```

Output:

Enter any String below [140] characters : VijayKumar
Reverse string :ramuKyajiV

(ii)

```
#include <string.h>
#include <string.h>
int main()
{
    char s[100];
    clrscr();
    printf("\n\nEnter a string below[100] characters to reverse\n");
    gets(s);
    strrev(s);
    printf("Reverse of the string: %s\n", s);
    getch();
    return 0;
}
```

Output:

Enter a string below[100] characters to reverse LORDS
Reverse of the string: SDROL

c. Write a C program to determine if the given string is a palindrome or not.

```
#include<stdio.h>
#include<conio.h>
int main()
{
    char string[140];
    int length=0, flag=1,i;
    clrscr();
    printf("\n\nEnter string below [120] characters:\n");
    gets(string);
    for(i=0;string[i]!='\0';i++)
    {
        length++;
    }
    for(i=0;i< length/2;i++)
    {
        if( string[i] != string[length-1-i] )
        {
            flag=0;
            break;
        }
    }
    if(flag==1)
    {
        printf("\n%s is PALINDROME",string);
    }
    else
    {
        printf("\n%s is NOT PALINDROME",string);
    }
    getch();
    return 0;
}
```

Output:

Enter string below [120] characters: RADAR

RADAR is PALINDROME

11. Programs on Structures and Unions

- a. Write a C program to accept and display student details(Rollno, Name, Branch, Address) using structures.

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
typedef struct
{
    long int rollnum;
    char name[30];
    char Branch[20];
    char Address[200];
}Student;
int main()
{
    int i;
    char ch;
    Student students;
    clrscr();
    printf("Enter any one Student Details \n \n");
    //Name
    printf("Name: ");
    scanf("%[^\n]s",students.name);
    //Roll Number
    printf("RollNumber: ");
    scanf("%ld",&students.rollnum);
    //Branch
    printf("Branch: ");
    scanf("%s",students.Branch);
    //Address
    printf("Address: ");
    scanf("%s",students.Address);
    //Displaying Students details
    printf("\n----- Displaying Students Details ----- \n");
    printf("Name \t: ");
    printf("%s \n",students.name);
    printf("roll number \t: ");
```

```
printf("%ld \n",students.rollnum);
printf("Branch \t: ");
printf("%s \n",students.Branch);
printf("Address \t: ");
printf("%s \n",students.Address);
getch();
return 0;
}
```

Output:

Enter any one Student Details

Name: Vijay

RollNumber:564

Branch: CSE

Address: Hyd

----- Displaying Students Details -----

Name: Vijay

roll number: 564

Branch: CSE

Addres : Hyd

b. Create a union containing 6 strings: name, home_ address, hostel_ address, city, state and zip. Write a C program to display your present address.

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
union details
{
    char name[120];
    char home_add[130];
    char hostel_add[130];
    char city[30];
    char state[20];
    long int pincode;
}a;
void main()
{
    char sname[100],home[250],hostel[200],city[50],state[50];
    long int pincode;
    clrscr();
    printf("\n enter student name : ");
    gets(sname);
    printf("\n enter student permanent home address : ");
    gets(home);
    printf("\n enter student hostel : ");
    gets(hostel);
    printf("\n enter city where you stay : ");
    gets(city);
    printf("\n enter state you belong : ");
    gets(state);
    printf("\n enter pincode where you belong : ");
    scanf("%ld",&pincode);
    printf("\n ***** Displaying the STUDENT DETAILS :***** \n");
    strcpy(a.name,sname);
    printf("\n %s",a.name);
    strcpy(a.home_add,home);
    printf("\n %s",a.home_add);
    strcpy(a.hostel_add,hostel);
    printf("\n %s",a.hostel_add);
    strcpy(a.city,city);
```

```
    printf("\n %s",a.city);
    strcpy(a.state,state);
    printf("\n %s",a.state);
    a.pincode=pincode;
    printf("\n %ld",a.pincode);
    getch();
}
```

Output:

```
enter student name : Vijay Kumar
enter student permanent home address : Flat no 32 Kukatpally
enter student hostel : Boys Hostel Mehdipatnam
enter city where you stay : Hyderabad
enter state you belong : Telangana
enter pincode where you belong : 500030
***** Displaying the STUDENT DETAILS :*****
Vijay Kumar
Flat no 32 Kukatpally
Mehdipatnam
Hyderabad
Telangana
500030
```

12. File Handling Programs

a. Write a C program to display the contents of a file.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    char ch, file_name[25];
    FILE *fp;
    clrscr();
    printf("Enter name of a file you wish to see the contents of it\n");
    gets(file_name);
    fp = fopen(file_name, "r"); // read mode
    if (fp == NULL)
    {
        perror("Error while opening the file.\n");
        exit(EXIT_FAILURE);
    }
    printf("The contents of %s file are:\n", file_name);
    while((ch = fgetc(fp)) != EOF)
    printf("%c", ch);
    fclose(fp);
    getch();
    return 0;
}
```

Output:

Enter name of a file you wish to see
one.c

The contents of one.c file are:

```
void main()
{
    clrscr();
    printf("Hello World");
    getch();
}
```

b. Write a C program to copy the contents of one file to another.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    char ch, source_file[30], target_file[30];
    FILE *source, *target;
    clrscr();
    printf("Enter name of file to copy\n");
    gets(source_file);
    source = fopen(source_file, "r");
    if( source == NULL )
    {
        printf("Press any key to exit...\n");
        exit(EXIT_FAILURE);
    }
    printf("Enter name of target file\n");
    gets(target_file);
    target = fopen(target_file, "w");
    if( target == NULL )
    {
        fclose(source);
        printf("Press any key to exit...\n");
        exit(EXIT_FAILURE);
    }
    while( ( ch = fgetc(source) ) != EOF )
        fputc(ch, target);
    printf("File copied successfully.\n");
    fclose(source);
    fclose(target);
    getch();
    return 0;
}
//Before executing try to create or give existing file for source file.
```

Output:

```
Enter name of file to copy
one.c
Enter name of target file
b1.txt
File copied successfully
```

c. Write a program to count no of lines, words and characters in a given text file.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    FILE * file;
    char path[100];
    char ch;
    int characters, words, lines;
    clrscr();
    /* Input path of files to merge to third file */
    printf("Enter source file path: ");
    scanf("%s", path);
    /* Open source files in 'r' mode */
    file = fopen(path, "r");
    /* Check if file opened successfully */
    if (file == NULL)
    {
        printf("\nUnable to open file.\n");
        printf("Please check if file exists and you have read privilege.\n");
        exit(EXIT_FAILURE);
    }
    /*Logic to count characters, words and lines.*/
    characters = words = lines = 0;
    while ((ch = fgetc(file)) != EOF)
    {
        characters++;
        /* Check new line */
        if (ch == '\n' || ch == '\0')
            lines++;
        /* Check words */
        if (ch == ' ' || ch == '\t' || ch == '\n' || ch == '\0')
            words++;
    }
    /* Increment words and lines for last word */
    if (characters > 0)
    {
        words++;
        lines++;
    }
    /* Print file statistics */
```

```
printf("\n");
printf("Total characters = %d\n", characters);
printf("Total words    = %d\n", words);
printf("Total lines    = %d\n", lines);
/* Close files to release resources */
fclose(file);
getch();
return 0;
}
```

//Any text file you can give while executing. Here i have given one.c file only

Output:

Enter source file path:

one.c

Total characters = 74

Total words = 9

Total lines = 7