# SASTRA Deemed University
# C-Programs
# I-Btech-2018


**I. Simple Programs -03**

1. Find area of a rectangle
2. Find ASCII value of a character
3. Convert Celsius to Fahrenheit
4. Swap value of two variables
5. Check the given number is odd or even
6. Check whether a character is vowel or consonant
7. Find largest among three numbers
8. Leap year checking
9. Positive negative checking
10. Quadratic equation

**II. Programs using Loops -05**

11. Factorial without using function
12. Table of N and square of N
13. Calculate x to the power y
14. Multiplication table
15. Sum of natural numbers
16. Fibonacci starting from any two numbers
17. Upper case to Lower case
18. Lower to upper
19. Pascal triangle
20. LCM & GCD
21. Prime numbers between two ranges
22. Factors of a number
23. Prime Factors
24. Bin to dec and oct
25. Count the number of digit in an integer
26. Reverse the digits of given number
27. Number palindrome
28. Digit summation
29. Amstrong checking
30. Make simple calculator in C
31. TO FIND SIN(X) USING SINE SERIES
32. TO FIND cos(X) USING cos SERIES
33. Exponent series
34. Floyds Triangle

III. **Programs using Arrays -09**

35. Fibonacci using array
36. Largest among N numbers in an array
37. Smallest among N numbers in an array
38. Reverse the array elements
39. Insert an element in an array
40. Deleting an array element
41. Transpose of a matrix
42. Duplication removal
43. Linear Search
44. Binary search
45. Split the sorted array
46. Matrix addition
47. Matrix multiplication
48. Inverse of a 3X3 matrix

IV. **VI. Programs using Functions --13**

49. Factorial using function
50. Min and Max of array
51. Bubble Sort
52. Convert :Bin to dec; dec to bin
53. Bin to oct; oct to bin
54. Dec to Hex
55. Oct to dec; dec to oct
56. Factorial using recursive function
57. Fibonacci using recursive function
58. Find the nth no. of a fibonacci series
59. Sum of N numbers using recursion
60. Reverse the sentence using recursion
61. Power using recursion
62. Towers of Hanoi
63. Exponent using recursion
64. GCD using recursion

**V Programs using Structures --18**

65. Student structure
66. Players structure
67. Add two polynomials using structures in function
68. Add two distances using structures
69. Add two complex numbers
70. Calculate difference between two time period

**VI Programs using Strings --21**

71. Program to Count Blanks,Tabs and Newlines
72. Palindrome checking
73. convert a name into its ascii values.
74. calculating string length without strlen function
75. comparing 2 strings without strcmp function
76. copying one string to another without using strcpy
77. string concatenation without using strcat function
78. Pattern replacement
79. Finding vowels
80. Sorting in alphabetical order
81. Searching substring in a string
82. Find the frequency of a character in a string
83. Remove characters in string except alphabets
84. Reverse the given string

**VII Programs Using pointers -25**

85. Area of circle using pointer
86. Function pointer
87. Duplication removal using pointer
88. Sorting integer array using pointer
89. Some of array using pointer
90. Count no. of words using pointer
91. Length of a string using pointer
92. Reverse the string using pointer

**VIII Miscellaneous -27**

93. Dec to Bin using bits
94. Find largest element using dynamic memory allocation
95. Matrix multiplication using dynamic memory allocation
96. Add the digits of the number using single statement
97. Reverse the digits without using % operator
98. Addition without + sign
99. Addition without using arithmetic operators
100. Stack operation
101. to 109. Printing Patterns
110. Sum of 2 matrix using dynamic memory allocation
111. Employee structure sorting based on salary
112. Merge sorting using dynamic memory allocation
113. Addition of quadratic equations using structures
114. Insertion Sort

## I. Simple Programs

1. Find area of rectangle

```
#include<stdio.h>
 int main()
{int length,breadth,area;
printf("\nEnter the Length of Rectangle : ");
scanf("%d",&length);
printf("\nEnter the Breadth of Rectangle : ");
scanf("%d",&breadth);
area = length * breadth;
printf("\nArea of Rectangle : %d",area);
}
```

2.Find ASCII value of a character

```
#include <stdio.h>
int main()
{  char c;
   printf("Enter a character: ");
   scanf("%c",&c);
   printf("ASCII value of %c = %d",c,c);
}
```

3.Convert Celsius to Fahrenheit

```
#include<stdio.h>
int main()
{float celsius,fahrenheit;
printf("\nEnter temp in Celsius : ");
scanf("%f",&celsius);
fahrenheit = (1.8 * celsius) + 32;
printf("\nTemperature in Fahrenheit : %f ",fahrenheit);
}
```

4.Swap the value of two variables

```
#include<stdio.h>
 int main()
{ float a, b, temp;
printf("Enter value of a: ");
scanf("%f",&a);
printf("Enter value of b: ");
scanf("%f",&b);
temp = a;
a = b;
b = temp;
printf("\nAfter swapping, value of a = %.2f\n", a);
printf("After swapping, value of b = %.2f", b); }
```

5.Check the given number is odd or even

```
#include <stdio.h>
int main()
{ int num;
printf("Enter an integer you want to check: ");
scanf("%d",&num);
if((num%2)==0)
printf("%d is even.",num);
else
printf("%d is odd.",num); }
```

6.Check whether a character is vowel or consonant

```
#include <stdio.h>
int main()
{ char c;
 printf("Enter an alphabet: ");
scanf("%c",&c);
if(c=='a'||c=='A'||c=='e'||c=='E'||c=='i'||c=='I'||c=='o'||c=='
O'||c=='u'||c=='U')
printf("%c is a vowel.",c);
 else
printf("%c is a consonant.",c);
 }
```

7. Find largest among three numbers

```
#include <stdio.h>
int main()
{ float a, b, c;
printf("Enter three numbers: ");
scanf("%f %f %f", &a, &b, &c);
if(a>=b && a>=c)
printf("Largest number = %.2f", a);
else if(b>=c)
printf("Largest number = %.2f", b);
else printf("Largest number = %.2f", c);
}
```

8. Leap year checking

```
#include<stdio.h>
 int main()
{ int year;
printf("Enter a year: ");
scanf("%d",&year);
if(year%4 == 0)
{ if( year%100 == 0) /* Checking for a century year */
{ if ( year%400 == 0)
printf("%d is a leap year.", year);
            else printf("%d is not a leap year.", year); }
 else printf("%d is a leap year.", year ); }
else printf("%d is not a leap year.", year);
}
```

9. Positive negative checking

```
#include <stdio.h>
int main()
{ float num;
 printf("Enter a number: ");
 scanf("%f",&num);
if (num<0)
printf("%.2f is negative.",num);
 else if (num>0)
```

```
printf("%.2f is positive.",num);
else printf("You entered zero."); }
```

10. Quadratic equation

```
#include<stdio.h>
#include<math.h>
main()
{int a,b,c;
float d,p,q;
printf("Values of a,b,c?");
scanf("%d%d%d",&a,&b,&c);
d=((b*b)-(4*a*c));
if(d>0)
{printf("real");
p=(-b+sqrt(d))/(2*a);
q=(-b-sqrt(d))/(2*a);
printf("the roots are %f %f",p,q);}
else if(d<0)
printf("imaginary");
else
{printf("real and equal");
p=((-b)+sqrt(d))/(2*a));
q=((-b)-sqrt(d))/(2*a));
printf("the roots are %f %f",p,q);}
}
```

## II. Programs using Loops

11. Factorial without using function

```
#include<stdio.h>
int main()
{int i,number,factorial;
printf("\nEnter the number : ");
scanf("%d",&n);
factorial = 1;
for(i=1;i<=n;i++)
     factorial = factorial * i;
printf("\nFactorial of %d is %d",n,factorial );
}
```

12. Table of N and square of N

```
#include<stdio.h>
int main()
{      int i,n;
Printf("Enter the value of n");
Scanf("%d",&n);
Printf("The value of n and square of it is");
for(i=1;i<=n;i++)
printf("%d\t%d\n",i,i*i);
}
```

13. Calculate x to the power y

```
#include <stdio.h>
int main()
 { int base, exp;
long long int value=1;
 printf("Enter base number and exponent ");
 scanf("%d%d", &base, &exp);
while (exp!=0)
{ value*=base;
--exp; }
printf("Answer = %d", value); }
```

14. Multiplication table

```
#include <stdio.h>
int main()
 { int n, i;
printf("Enter an integer to find multiplication table: ");
scanf("%d",&n);
for(i=1;i<=20;++i)
printf("%d * %d = %d\n", n, i, n*i);
  }
```

15. Sum of natural numbers

```
#include <stdio.h>
int main()
 { int n, count, sum=0;
printf("Enter an integer: ");
scanf("%d",&n);
count=1;
while(count<=n)
 { sum+=count;
++count; }
printf("Sum = %d",sum);
  }
```

16. Fibonacci starting from any two numbers

```
#include<stdio.h>
 int main()
{int first,second,sum,num,counter=0;
printf("Enter the term : ");
scanf("%d",&num);
printf("\nEnter First Number : ");
scanf("%d",&first);
printf("\nEnter Second Number : ");
scanf("%d",&second);
printf("\nFibonacci Series : %d  %d  ",first,second);
while(counter< num)
   {    sum=first+second;
   printf("%d  ",sum);
   first=second;
   second=sum;
   counter++;    }
}
```

17. Uppercase to Lower case

```
#include<stdio.h>
int main()
{  char str[20];
  int i=0,sl=0;
  printf("Enter any string->");
  scanf("%s",str);
  printf("The string is->%s",str);
  while(str[i]!='\0')
{
  i++;
  sl++;
}
  for(i=0;i<sl;i++){
     if(str[i]>=65&&str[i]<=90)
      str[i]=str[i]+32;  }
  printf("\nThe string in lower case is->%s",str);
  }
```

18. Lower to Upper

```
#include<stdio.h>
int main()
{  char str[20];
  int i=0,sl=0;
  printf("Enter any string->");
```

```
 scanf("%s",str);
 printf("The string is->%s",str);
while(str[i]!='\0')
{
 i++;
 sl++;
}
 for(i=0;i<sl;i++){
        if(str[i]>=97&&str[i]<=122)
        str[i]=str[i]-32;  }
 printf("\nThe string in lowercase is->%s",str);
 }
```

19. Pascal triangle

```
#include<stdio.h>
int main()
{int bin=1,p,q=0,r,x;
printf("Rows you want to input:");
scanf("%d",&r);
printf("\n Pascal's Triangle:\n");
while(q<r)
{for(p=40-3*q;p>0;--p)
printf(" ");
for(x=0;x<=q;++x)
{if((x==0)||(q==0))
bin=1;
else
bin=(bin*(q-x+1))/x;
printf("%6d",bin);
}
printf("\n");
++q;}
}
```

O/P: For r=4:

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

20. LCM & GCD

```
# include <stdio.h>
int main()
{int n1, n2, prod, gcd, lcm,m,i ;
printf("Enter the two numbers : ") ;
scanf("%d %d", &n1, &n2) ;
prod = n1 * n2 ;
if(n1>n2 )
      m=n2;
  else
      m=n1;
       for(i=m;i>=1;i--){
       if(n1%i==0 && n2%i==0){
          gcd = i ;
          break;      }   }
lcm = prod / gcd ;
printf("\nThe GCD is : %d", gcd) ;
printf("\n\nThe LCM is : %d", lcm);
}
```

21. Prime numbers between two ranges

```
#include <stdio.h>
int main()
{ int n1, n2, i, j, flag;
printf("Enter two numbers(intevals): ");
scanf("%d %d", &n1, &n2);
printf("Prime nos in range %d - %d are: ", n1, n2);
for(i=n1+1; i<n2;++i)
{flag=0;
for(j=2;j<=i/2;++j)
{   if(i%j==0)
     {flag=1;
     break;}}
     if((flag==0)&&(i!=1))
     printf("%d  ",i);}
}
```

22. Factors of a number

```
#include <stdio.h>
int main()
 { int n,i;
printf("Enter a positive integer: ");
 scanf("%d",&n);
 printf("Factors of %d are: ", n);
for(i=1;i<=n;++i)
{ if(n%i==0)
printf("%d  ",i); }
 }
```

23. Prime Factors

```
 #include<stdio.h>
int main()
{  int n,i;
printf("Enter a Number:");
scanf("%d",&n);
printf("\n\nPrime Factors of %d is: ",n);
for(i=2;i<=n;i++)
{   if(n%i==0)
      {   printf("%d,",i);
       n=n/i;
       i--;
       if(n==1)
break;   }   }
}
```

24. Dec to Bin and oct

```
#include<stdio.h>
int main()
{long int
decNum,quotient,binNum=0,pos=1,octnum=0,quot;
printf("Enter any decimal number: ");
scanf("%ld",& decNum);
quotient = decNum
quot= decNum;
while(quotient!=0)
{  binNum= binNum+ pos*(quotient % 2);
 quotient = quotient / 2;
 pos=pos*10;}
pos=1;
printf("Binary equivalent of decimal number %ld is
%ld\n", decNum,binNum);
while(quot!=0)
{  octnum= octnum+ pos*(quot % 8);
 quot = quot / 8;
 pos=pos*10;}
printf("octal equivalent of decimal number %ld is %ld",
decNum,octnum);
}
```

25. Count the number of digit in an integer

```
#include <stdio.h>
int main()
 { int n,count=0;
printf("Enter an integer: ");
scanf("%d", &n);
do
{ n/=10;
++count;
}while(n!=0);
printf("\number of digits: %d",count); }
```

26. Reverse the digits of given number

```
#include<stdio.h>
int main()
{long int num,r,sum=0,giv;
   printf("Enter a number: ");
   scanf("%ld",&num);
    giv=num;
     while(num){
     r=num%10;
     num=num/10;
     sum=sum*10+r;    }
    printf("reverse of given number %ld is
%ld",giv,sum);
}
```

27. .Number palindrome

```
#include<stdio.h>
int main()
{long int num,r,sum=0,temp;
   printf("Enter a number: ");
   scanf("%ld",&num);
   temp=num;
   while(num)  {
       r=num%10;
       num=num/10;
       sum=sum*10+r;    }
   if(temp==sum)
            printf("%ld is a palindrome",temp);
   else
            printf("%ld is not a palindrome",temp);
}
```

28. Digit summation

```
# include<stdio.h>
 int main()
{int sum=0,m,n;
printf("enter the value of n");
scanf("%d",&n);
while(n!=0)
{m=n%10;
n=n/10;
sum=sum+m;    }
printf("the value is %d",sum);
}
```

28. Amstrong checking

```
#include<stdio.h>
int main()
{long int num,r,sum=0,ams;
   printf("Enter a number: ");
  scanf("%ld",&num);
  ams=num;
   while(num){
       r=num%10;
       num=num/10;
       sum=sum+r*r*r;    }
   if(ams==sum)
      printf("%ld is a amstrong number",ams);
   else
      printf("%ld is not a amstrong number ",ams);
}
```

29. Making simple calculator inC

```
#include <stdio.h>
int main()
{ char operator;
```

```
float num1,num2;
printf("Enter operator either + or - or * or divide : ");
scanf("%c",&operator);
printf("Enter two operands: ");
scanf("%f%f",&num1,&num2);
switch(operator)
{case '+': printf("\n num1+num2=%f",num1+num2);
break;
case '-': printf("\n num1-num2=%f",num1-num2);
break;
case '*': printf("\n num1*num2=%f",num1*num2);
break;
case '/': printf("\n num2/num1 = %f",num1/num2);
break;
default: printf("\n Error! operator is not correct"); break;
}
 }
```

30. Find sin(x) using series

```
#include<stdio.h>
 #include<math.h>
int main()
{float sum,term,xd,x;
int i;
printf("Enter x in degree:");
scanf("%f",&xd);
x=(xd*3.141552654)/180.0;
sum=0;
term=x;
for(i=2;fabs(term)>0.000001;i++)//fabs(x)- returns
modulus i.e. absolute value of argument(x)
{sum+=term;
term=-term*x*x/((2*i-1)*(2*i-2));}
printf("Sin (%f)=%f",xd,sum);
}
```

31. Cos(x) series

```
#include<stdio.h>
int main()
{   int i, n;
   float x, sum=1, t=1;
    printf(" Enter the value for x : ");
   scanf("%f",&x);
   printf(" Enter the value for n : ");
   scanf("%d",&n);
    x=x*3.14159/180;
     /* Loop to calculate the value of Cosine */
   for(i=1;i<=n;i++)
   {
      t=t*(-1)*x*x/(2*i*(2*i-1));
      sum=sum+t;
   }
    printf(" The value of Cos(%f) is : %.4f", x, sum);
   }
```

32. Exponent series

```
#include<stdio.h>
#define ACCURACY 0.0001
int main()
{ int n, count;
float x, term, sum;
 printf("Enter value of x:");
 scanf("%f", &x);
 n = term = sum = count = 1;
while (n <= 100)
    { term = term * x/n;
    sum = sum + term;
    count = count + 1;
if (term < ACCURACY)
        n = 999;
else
        n = n + 1;    }
 printf("Terms = %d Sum = %f\n", count, sum);
 }
```

33. FLOYD'S TRIANGLE

```
#include<stdio.h>
int main()
{  int i,j,k=1;
int range;
  printf("Enter the range: ");
  scanf("%d",&range);
  printf("FLOYD'S TRIANGLE : n \n");
for(i=1;i<=range;i++)
     {     for(j=1;j<=i;j++,k++)
           printf("%d  ",k);
     printf("\n");      }
  }
```

FLOYD'S TRIANGLE : for *range=4*

```
1
2 3
4 5 6
7 8 9 10
```

## III. Programs using Arrays

34. Fibonacci using array

```
#include<stdio.h>
 main()
{int n,fib[25];
scanf("%d",&n);
fib[0]=0;
fib[1]=1;
for(i=2;i<=n;i++)
fib[i]=fib[i-2]+fib[i-1];
for(i=0;i<=n;i++)
printf("%d\n",fib[i]);
}
```

35. Largest among N numbers in an array

```
#include<stdio.h>
int main()
  {  int a[30],i,n,largest;
  printf("\n Enter no of elements :");
  scanf("%d",&n);
for(i=0 ; i < n ; i++)
    scanf("%d",&a[i]);
  largest = a[0];
for(i = 0;i<n;i++)
  {  if(a[i] > largest )
     largest = a[i];  }
    printf("\nLargest Element : %d",largest)
}
```

36. Smallest among N numbers in an array

```
#include<stdio.h>
int main()
{ int a[30],i,n,smallest;
 printf("\n Enter no of elements :");
 scanf("%d",&n);
for(i=0 ; i < n ; i++)
   scanf("%d",&a[i]);
  smallest = a[0];
for(i = 0 ;i < n ; i++)
  {  if ( a[i] < smallest )
   smallest = a[i];  }
printf("\nSmallest Element : %d",smallest);
   }
```

37. Reverse the array elements

```
#include<stdio.h>
int main()
{ int a[30],i,j,n,temp;
 printf("\n Enter no of elements :");
 scanf("%d",&n);
for(i=0 ; i < n ; i++)
 scanf("%d",&a[i]);
 j = i-1;   // j will Point to last Element
 i = 0;     // i will be pointing to first element
while(i < j)
 { temp = a[i];
 a[i] = a[j];
 a[j] = temp;
 i++;            // increment i and decrement j
 j--; }
for(i = 0 ;i< n ;i++)
  printf("\n %d",a[i]);   }
```

38. Insert an element in an array

```
#include<stdio.h>
int main()
{ int arr[30],element,num,i,location;
 printf("\n Enter no of elements :");
 scanf("%d",&num);
 for(i=0 ; i < num ; i++)
            scanf("%d",&arr[i]);
 printf("\n Enter the element to be inserted :");
 scanf("%d",&element);
 printf("\n Enter the location");
 scanf("%d",&location);
  for(i = num ;i >= location ; i--)
     arr[i] = arr[i-1];
 num++;
 arr[location-1] = element;
for(i = 0 ;i < num ;i++)
       printf("\n %d",arr[i]);
}
```

39. Deleting an array element

```
#include<stdio.h>
int main()
{ int a[30],n,i,j;
 printf("\n Enter no of elements :");
 scanf("%d",&n);
printf("\n Enter %d elements :",n);
for(i=0;i < n;i++)
 scanf("%d",&a[i]);
printf("\n location of the element to be deleted :");
 scanf("%d",&j);
while(j < n)
 {  a[j-1]=a[j];
  j++; }
 n--;
for(i=0;i < n;i++)
printf("\n %d",a[i]);
}
```

40. Transpose of a matrix

```
#include<stdio.h>
int main()
{ int a[10][10],m,i,j,temp;
printf(“\n Enter the size of matrix :");
 scanf("%d",&m);
printf(“\n Enter the values a:");
for(i=0;i<m;i++)
for(j=0;j<m;j++)
    scanf("%d",&a[i][j]);
printf(“\nGiven square matrix is");
for(i=0;i<m;i++)
  {  printf(“\n");
for(j=0；j 〈 m；j++)
        printf("%d\t",a[i][j]); }
for(i=1;i<m;i++)
for(j=0;j<i;j++)
   {   temp=a[i][j];
   a[i][j]=a[j][i];
   a[j][i]=temp;   }
(“\nTranspose matrix is :");
for(i=0;i 〈 m;i++)
 {  printf(“\n");
for(j=0;j<m;j++)
     printf("%d\t",a[i][j]); }
}
```

41. Duplicate removal in an array

```
#include<stdio.h>
int main()
{ int a[50], i,j,k,size,n,t;
  printf("\nEnter size of the array: ");
  scanf("%d",&n);
  printf("\nEnter %d elements into the array: ",n);
  for(i=0;i<n;i++)
    scanf("%d",&a[i]);
  size=n;
  for(i=0;i<size;i++){
    for(j=0;j<size;j++){
        if(i==j)       continue;
        else if(a[i]==a[j]){
            k=j;
            size--;
            while(k < size){
                a[k]=a[k+1];
                k++;            }
            j=0;             }      }   }
for(i=0;i<size;i++){
             for(j=i+1;j<size;j++){
             if(a[i]>a[j])
             {t=a[i];
             a[i]=a[j];
             a[j]=t; }     }   }
 printf("\nThe array after removing duplicates is: ");
 for(i=0;i < size;i++)
             printf(" %d ",a[i]);
}
```

42. Linear Search

```
#include<stdio.h>
int main()
{ int a[30],x,n,i;
printf(“\nEnter no of elements :");
scanf("%d",&n);
printf(“\nEnter the values :");
for(i=0;i < n;i++)
scanf("%d",&a[i]);
printf(“\nEnter the elements to be searched");
scanf("%d",&x);
i=0;
while(i < n && x!=a[i])
i++;
if(i < n) /* Element is found */
printf("found at the location =%d",i+1);
else
printf(“\n not found");
}
```

43. Binary search

```
#include<stdio.h>
 int main()
{int array[10];
int i, j, N, temp, keynum;
int low,mid,high;
printf("Enter the value of N\n");
scanf("%d",&N);
printf("Enter the elements one by one\n");
for(i=0;i<=N;i++) {
scanf("%d",&array[i]);}
printf("Enter the element to be searched\n");
scanf("%d", &keynum);
low=1;
high=N;
do
{mid= (low + high) / 2;
if ( keynum < array[mid] )
high = mid - 1;
else if ( keynum > array[mid])
low = mid + 1;
} while( keynum!=array[mid] && low <= high);
if( keynum == array[mid] )
```

```
printf("SUCCESSFUL SEARCH\n");
else
printf("Search is FAILED\n");
}
```

44. Split the sorted array

```
#include<stdio.h>
 int main()
{int array[10],les[10],big[10];
int i, j, N, flag, keynum;
printf("Enter the value of N\n");
scanf("%d",&N);
printf("Enter the elements one by one\n");
for(i=0;i<N;i++) {
scanf("%d",&array[i]);}
printf("Enter the sorted elements \n");
scanf("%d", &keynum);
for(i=0;i<N;i++)
{if ( keynum ==array[i] )
keynum=i;
flag=1;}
if( flag == 1 )
{printf("array created :smaller than number\n");
for(i=0;i<keynum;i++)
{les[i]=array[i];
printf("%d  ",les[i]);}
printf("array created :bigger than number\n");
for(i=keynum+1;i<N;i++)
{big[i]=array[i];
printf("%d  ",big[i]);}
}
else printf("give correct number\n");
}
```

45. Matrix addition

```
#include<stdio.h>
int main()
{
 int i,j,a[10][10],b[10][10],c[10][10],m1,n1,m2,n2;
printf("\nEnter the number of Rows of Mat1 : ");
scanf ("%d",&m1);
printf("\nEnter the number of Columns of Mat1 : ");
scanf ("%d",&n1);
for(i=0;i<m1;i++)
for(j=0;j<n1;j++)
   {    printf("Enter the Element a[%d][%d] : ",i,j);
   scanf("%d",&a[i][j]);    }
printf("\nEnter the number of Rows of Mat2 : ");
scanf ("%d",&m2);
printf("\nEnter the number of Columns of Mat2 : ");
scanf ("%d",&n2);
if ( m1 != m2 || n1 != n2 )
 { printf("\nOrder of two matrices is not same ");
 exit(0); }
for(i=0;i<m2;i++)
for(j=0;j<n2;j++)
   { printf("Enter the Element b[%d][%d] : ",i,j);
   scanf("%d",&b[i][j]);    }
for(i=0;i<m1;i++)
for(j=0;j<n1;j++)
       c[i][j] = a[i][j] + b[i][j] ;
printf("\nThe Addition of two Matrices is : \n");
for(i=0;i<m1;i++)
 {    for(j=0;j<n1;j++ )
        printf("%d\t",c[i][j]);
  printf("\n"); }
}
```

46. Matrix multiplication

```
#include <stdio.h>
 int main()
{  int m, n, p, q, i,j, k, sum = 0;
  int first[10][10], second[10][10], multiply[10][10];
 printf("Enter number of rows and columns of first
matrix\n");
  scanf("%d%d", &m, &n);
  printf("Enter elements of first matrix\n");
   for (i = 0; i < m; i++)
    for (j = 0; j < n; j++)
      scanf("%d", &first[i][j]);
  printf("Enter number of rows and columns of second
matrix\n");
 scanf("%d%d", &p, &q);

  if (n != p)
    printf("The matrices can't be multiplied with each
            other.\n");
  else
  {
    printf("Enter elements of second matrix\n");
     for (i = 0; i < p; i++)
      for (j = 0; j < q; j++)
        scanf("%d", &second[i][j]);

    for (i = 0; i < m; i++)
     { for (j = 0; j < q; j++)
      {multiply[i][j]=0;
        for (k = 0; k < p; k++)
          multiply[i][j] = multiply[i][j]  +
              first[i][k]*second[k][j];
         }
     }
```

```
    printf("Product of the matrices:\n");
    for (i = 0; i < m; i++)
  {  for (j = 0; j < q; j++)
       printf("%d\t", multiply[i][j]);
      printf("\n");
    }
  }
}
```

47. Inverse of a 3X3 matrix

```
#include<stdio.h>
void reduction(float a[][6],int size,int pivot ,int col)
{int i,j;
float factor;
  factor=a[pivot][col];
for(i=0;i<2*size;i++)
      a[pivot][i]/=factor;
for(i=0;i<size;i++)
if(i!=pivot)
     {  factor=a[i][col];
for(j=0;j<2*size;j++)
             a[i][j]=a[i][j]-a[pivot][j]*factor;      }
}
int main()
{float a[3][6];
int i,j;
for(i=0;i<3;i++)    // Append Unit Matrix
for(j=0;j<6;j++)
   {    if(j==i+3)
      a[i][j]=1;
else
      a[i][j]=0;   }
printf("\n Enter a 3 X 3 Matrix");
for(i=0;i<3;i++)
for(j=0;j<3;j++)
   scanf("%f",&a[i][j]);
for(i=0;i<3;i++)
   reduction(a,3,i,i);
printf("\nInvers Matrix");
for(i=0;i<3;i++)
 { printf("\n");
for(j=0;j<3;j++)
   printf("%8.3f  ",a[i][j+3]); }
}
```

## IV. Programs Using Function

49. Factorial using function

```
#include<stdio.h>
int findFactorial(int);
int main()
{  int i,factorial,num;
  printf("Enter a number: ");
  scanf("%d",&num);
  factorial = findFactorial(num);
  printf("Factorial of %d is: %d",num,factorial);
  return 0;
}
int findFactorial(int num)
{  int i,f=1;
    for(i=1;i<=num;i++)
      f=f*i;
    return f;
}
```

50. Find minimum number in an array

```
#include <stdio.h>
int minimum (int values[], int numberOfElements)
{int minValue, i;
minValue = values[0];
for ( i = 1; i < numberOfElements; ++i )
if ( values[i] < minValue )
minValue = values[i];
return minValue;
}
int main ()
{int array1[5] = { 157, -28, -37, 26, 10 };
int array2[7] = { 12, 45, 1, 10, 5, 3, 22 };
printf ("array1 minimum: %i\n", minimum (array1, 5));
printf ("array2 minimum: %i\n", minimum (array2, 7));
}
```

51. Bubble Sort

```
#include<stdio.h>
void bubble_sort(int [],int);
int main()
{   int a[30],n,i;
    printf("\nEnter no of elements :");
    scanf("%d",&n);
    printf("\nEnter array elements :");
for(i=0;i<n;i++)
     scanf("%d",&a[i]);
    bubble_sort(a,n);
    }
void bubble_sort(int a[],int n)
{int i,j,k,temp;
  printf("\nUnsorted Data:");
for(k=0;k<n;k++)
       printf("%5d",a[k]);
for(i=1;i< n;i++)
   {       for(j=0;j< n-1;j++)
if(a[j]>a[j+1])
            { temp=a[j];
            a[j]=a[j+1];
            a[j+1]=temp;    }
   printf("\nAfter pass %d : ",i);
for(k=0;k< n;k++)
          printf("%5d",a[k]);   }
}
```

52. Convert :Bin to dec; dec to bin

```
#include<stdio.h>
#include<math.h>
int binary_decimal(int n);
int decimal_binary(int n);
int main()
{ int n; char c;
printf("1. Enter alphabet 'd' to convert binary to
decimal.\n");
 printf("2. Enter alphabet 'b' to convert decimal to
binary.\n");
scanf("%c",&c);
 if (c =='d' || c == 'D')
{ printf("Enter a binary number: ");
scanf("%d", &n);
printf("%d in binary = %d in decimal", n,
binary_decimal(n)); }
 if (c =='b' || c == 'B')
{ printf("Enter a decimal number: ");
 scanf("%d", &n);
 printf("%d in decimal = %d in binary", n,
decimal_binary(n)); }
 }
int decimal_binary(int n)
{ int rem, i=1, binary=0;
while (n!=0)
{ rem=n%2;
 n/=2;
binary+=rem*i;
i*=10; }
return binary; }
int binary_decimal(int n)
{ int decimal=0, i=0, rem;
while (n!=0)
{ rem = n%10;
n/=10;
decimal += rem*pow(2,i);
```

```
++i; }
return decimal;
}
```

53. bin to oct and oct to binary

```
#include<stdio.h>
#include<math.h>
int binary_octal(int n);
int octal_binary(int n);
int main()
 { int n; char c;
printf("Instructions:\n");
printf("Enter alphabet 'o' to convert binary to octal.\n");
printf("2. Enter alphabet 'b' to convert octal to
binary.\n");
scanf("%c",&c);
if ( c=='o' || c=='O')
 { printf("Enter a binary number: ");
 scanf("%d",&n);
printf("%d in binary = %d in octal", n, binary_octal(n));
}
if ( c=='b' || c=='B')
{ printf("Enter a octal number: ");
 scanf("%d",&n);
printf("%d in octal = %d in binary",n, octal_binary(n));
}
 }
int binary_octal(int n)
{ int octal=0, decimal=0, i=0;
while(n!=0)
{ decimal+=(n%10)*pow(2,i);
 ++i;
n/=10; }
i=1;
while (decimal!=0)
{ octal+=(decimal%8)*i;
 decimal/=8; i*=10; }
return octal; }
int octal_binary(int n)
{ int decimal=0, binary=0, i=0;
while (n!=0)
{ decimal+=(n%10)*pow(8,i);
++i;
n/=10; }
 i=1;
while(decimal!=0)
{ binary+=(decimal%2)*i;
decimal/=2;
i*=10; }
return binary; }
```

54. dec to hex

```
#include<stdio.h>
#include<math.h>
void dec_hex(longint num)   // Function Definition
{longint rem[50],i=0,length=0;
while(num>0)
   { rem[i]=num%16;
     num=num/16;
     i++;
     length++;   }
printf("Hexadecimal number : ");
for(i=length-1;i>=0;i--)
  {    switch(rem[i])
    {      case 10:
           printf("A");
break;
case 11:
           printf("B");
break;
case 12:
           printf("C");
break;
case 13:
           printf("D");
break;
case 14:
           printf("E");
break;
case 15:
           printf("F");
break;
default :
          printf("%ld  ",rem[i]);    }
  }}
int main()
{
longint num;
printf("Enter the decimal number : ");
 scanf("%ld",&num);
    dec_hex(num);
}
```

55. dec to octal

```
#include<stdio.h>
#include<math.h>
int decimal_octal(int n);
int octal_decimal(int n);
int main()
{ int n; char c;
printf("Instructions:\n");
printf("1. Enter alphabet 'o' to convert decimal to
octal.\n");
```

```
printf("2. Enter alphabet 'd' to convert octal to
decimal.\n");
scanf("%c",&c);
if (c =='d' || c == 'D')
{ printf("Enter an octal number: ");
scanf("%d", &n);
printf("%d in octal = %d in decimal", n,
octal_decimal(n)); }
 if (c =='o' || c == 'O')
 { printf("Enter a decimal number: ");
scanf("%d", &n);
printf("%d in decimal = %d in octal", n,
decimal_octal(n)); }
 }
int decimal_octal(int n)
 { int rem, i=1, octal=0;
 while (n!=0)
{ rem=n%8;
n/=8;
 octal+=rem*i;
i*=10; }
return octal; }
int octal_decimal(int n)
{ int decimal=0, i=0, rem;
while (n!=0)
{ rem = n%10;
n/=10;
decimal += rem*pow(8,i);
++i; }
return decimal;
}
```

56. Factorial using recursive function

```
#include<stdio.h>
int main()
{int n,x,i,a;
int factorial(int);
printf("any number\n");
scanf("%d",&n);
x=factorial(n);
printf("the factorial of %d is %d",n,x);
}
int factorial(int n)
{if(n==1)
return (1);
else
return(n*factorial(n-1));
}
```

57. Fibonacci using recursive function

```
#include<stdio.h>
 fib(int,int,int);
int main()
{int n;
scanf("%d",&n);
fib(n,0,1);
}
fib(int n,int a,int b)
{int c;
c=a+b;
printf("\n%d",c);
n--;
if(n==0)
return;
fib(n,b,c);
}
```

58. Find the nth Number in Fibonacci series

```
#include <stdio.h>
int fibo(int);
int main()
{    int num;
   int result;
    printf("Enter the nth number in fibonacci series: ");
   scanf("%d", &num);
   if (num < 0)
   {  printf("Fibonacci of negative number is not
   possible.\n");
   }
   else
   {  result = fibo(num);
      printf("The %d number in fibonacci series is
   %d\n", num, result);
   }
}
int fibo(int num)
{
   if (num == 0)
        return 0;
   else if (num == 1)
   return 1;
   else
return(fibo(num - 1) + fibo(num - 2));
   }
```

59. Sum of N numbers using recursion

```
#include<stdio.h>
int add(int n);
int main()
```

```
{ int n;
printf("Enter an positive integer: ");
scanf("%d",&n);
printf("Sum = %d",add(n));  }
int add(int n)
{ if(n!=0)
return  n+add(n-1);
}
```

60. Reverse the sentence using recursion

```
#include<stdio.h>
void Reverse();
int main()
{ printf("Enter a sentence: ");
Reverse();
 }
void Reverse()
{ char c;
scanf("%c",&c);
if( c != '\n')
{ Reverse();
printf("%c",c);
}
}
```

61. Power using recursion

```
#include<stdio.h>
int power(int n1,int n2);
int main()
{ int base, exp;
printf("Enter base number: ");
scanf("%d",&base);
printf("Enter power number(positive integer): ");
scanf("%d",&exp);
printf("%d^%d = %d", base, exp, power(base, exp));
 }
int power(int base,int exp)
{ if ( exp!=1 )
return (base*power(base,exp-1));
else
 return base;
}
```

62. tower of hanoi

```
#include<stdio.h>
void towers(int,char,char,char);
void towers(int n,char frompeg,char topeg,char auxpeg)
{ /* If only 1 disk, make the move and return */
 if(n==1)
  { printf("\nMove disk 1 from peg %c to peg
%c",frompeg,topeg);
                return;
}
/* Move top n-1 disks from A to B, using C as auxiliary
*/
  towers(n-1,frompeg,auxpeg,topeg);
/* Move remaining disks from A to C */
printf("\nMove disk %d from peg %c to peg
%c",n,frompeg,topeg);
 /* Move n-1 disks from B to C using A as auxiliary */
              towers(n-1,auxpeg,topeg,frompeg);
             }
int main()
{ int n;
 printf("Enter the number of disks : ");
 scanf("%d",&n);
  printf("The Tower of Hanoi involves the moves
:\n\n");
  towers(n,'A','C','B');
}
```

ANOTHER WAY

```
#include<stdio.h>
void tower(int n, char a, char b, char c)
{if(n>=1)
 {tower(n-1,a,c,b);
  printf("\n Move disk from %c to %c",a,c);
tower(n-1,b,a,c);
}
}
int main()
{int n,i,step=1;
printf("enter number of disk\n");
scanf("%d",&n);
tower(n,'A','B','C');
for(i=1;i<=n;i++)
step=step*2;
printf("\nthe number of steps used is %d",step-1);
}
```

63. exponent using recurssion

```
#include<stdio.h>
int exp_rec(int,int);
int main()
{int n1,n2,res;
scanf("%d%d",&n1,&n2);
res=exp_rec(n1,n2);
printf("\n%d",res);
}
int exp_rec(int x, int y)
{if(y==0)
            return 1;
```

```
else
        return(x*exp_rec(x,y-1));
}
```

64. GCD

```
#include<stdio.h>
int GCD(int,int);
int main()
{
int n1, n2, res;
scanf("%d%d",&n1,&n2);
res=GCD(n1,n2);
printf("gcd=%d",res);
}
int GCD(int x,int y)
{int rem;
rem=x%y;
if(rem==0)
return y;
else
return(GCD(y,rem));
}
```

## V. Programs Using Structure

65.Student structure

```
#include<stdio.h>
 struct stu
{char name[25];
int rno;
int m[5];
struct date
{   int d,m,y;
   }
   dob;
   }s[20];
void main()
{  int total,tot,n,i,j;
  float avg,avgs;
  printf("\nenter the no of student ");
  scanf("%d",&n);
  for(i=0;i<n;i++)
  {  printf("\nname,date,no\n");
scanf("%s%d%d%d%d",s[i].name,&s[i].dob.d,&s[i].do
b.m,&s[i].dob.y,&s[i].rno);
  printf("enter the marks1-5\n");
  total=0;
  for(j=0;j<5;j++)
  {  printf("marks-%d \t",j+1);
  scanf("%d",&s[i].m[j]);
  total+=s[i].m[j];  }
  printf("total \t %d",total);
  avg=total/5.00;
  printf("\navg marks of student is %f\n",avg);  }
  for(i=0;i<5;i++)
  {  tot=0;
  for(j=0;j<=n;j++)
  {  tot=tot+s[j].m[i];  }
  avgs=tot/n;
  printf("sub:%d \n avg%f\n",i+1,avgs);    }
}
```

66.Players detail- structure

```
#include<stdio.h>
 struct play
{char name[25];
int age;
int nmatch;
int run;
float avgrun;
}cri[100],t;
void main()
{  int n,i,j;
  float d;
  printf("\nEnter the no of players ");
  scanf("%d",&n);
  for(i=0;i<n;i++)
  {  printf("\nEnter name,age,no of matches,total
runs\n");

scanf("%s%d%d%d",cri[i].name,&cri[i].age,&cri[i].nm
atch,&cri[i].run);  }
  for(i=0;i<n;i++)
          cri[i].avgrun=cri[i].run/cri[i].nmatch;
   for(i=0;i<n;i++)
          for(j=0;j<n-i-1;j++)
   {   if(cri[j].avgrun>cri[j+1].avgrun)
              {t=cri[j+1];
                    cri[j+1]=cri[j];
                    cri[j]=t;}
    }
printf("\ndetails in ascending order\n");
printf("\nName\tage\tmatches\truns\tavg_run");
    for(i=0;i<n;i++)

   printf("\n%s\t%d\t%d\t%d\t%f",cri[i].name,cri[i].age,
   cri[i].nmatch,cri[i].run,cri[i].avgrun);
    }
```

67.Addition of polynomial using structure in function

```
#include<stdio.h>
 #define MAX 20
 struct addpolynomial {
 int exp, coef;
};
//function to read polynomial
int read_addpolynomial(struct addpolynomial p[]) {
 int i, texp;
 i = 0;
 printf("\nEnter exp ( use -1 to exit) : ");
 scanf("%d", &texp);
 while (texp != -1) {
  p[i].exp = texp;
  printf("\nEnter coef : ");
  scanf("%d", &p[i].coef);
  i++;
  printf("\nEnter exp ( use -1 to exit) : ");
  scanf("%d", &texp); }
 return (i);}
 //function to print polynomial
int print_addpolynomial(struct addpolynomial p[], int
max1) {
 int i;
 for (i = 0; i < max1; i++)
  printf("%+dX%d  ", p[i].coef, p[i].exp);
```

```
 return;
}
 //function to ad polynomials
int add_addpolynomial( p1, p2, p3, max1, max2)
struct addpolynomial p1[], p2[], p3[];
int max1, max2;
{ int i,j,k;
 i = j = k = 0;
 while ( i <max1 && j <max2)
 { if( p1[i].exp > p2[j].exp)
  {   p3[k] = p1[i];
   k++;
   i++;  }
  else
  if( p1[i].exp < p2[j].exp)
  {   p3[k] = p2[j];
   k++;
   j++;  }
  else
  {   p3[k].exp = p1[i].exp;
   p3[k].coef = p1[i].coef + p2[j].coef;
   i++;
   j++;
   k++;  }
 }
 while( i <max1 )
 {  p3[k] = p1[i];
  k++;
  i++; }
 while( j <max2 )
 {  p3[k] = p2[j];
  k++;
  j++;
 }
 return(k);
}
void main() {
 struct addpolynomial p1[MAX], p2[MAX], p3[MAX];
 int max1, max2, max3;
 printf("\nEnter first addpolynomial : ");
 max1 = read_addpolynomial(p1);
 printf("\nEnter second addpolynomial : ");
 max2 = read_addpolynomial(p2);
 max3 = add_addpolynomial(p1, p2, p3, max1, max2);
 printf("\nFirst addpolynomial is ");
 print_addpolynomial(p1, max1);
 printf("\nSecond addpolynomial is ");
 print_addpolynomial(p2, max2);
 printf("\n The resultant addpolynomial after addition
is");
 print_addpolynomial(p3, max3);
}
```

68.Add two distance using structure

```
#include <stdio.h>
struct Distance
{ int feet; float inch;
 }d1,d2,sum;
void main()
{ printf("Enter information for 1st distance\n");
printf("Enter feet: "); scanf("%d",&d1.feet);
 printf("Enter inch: "); scanf("%f",&d1.inch);
printf("\nEnter infromation for 2nd distance\n");
printf("Enter feet: "); scanf("%d",&d2.feet);
printf("Enter inch: "); scanf("%f",&d2.inch);
sum.feet=d1.feet+d2.feet;
sum.inch=d1.inch+d2.inch; /* If inch is greater than 12,
changing it to feet. */
 if (sum.inch>12.0)
{ sum.inch=sum.inch-12.0;
++sum.feet; }
printf("\nSum of distances=%d\'%.1f\"",
sum.feet,sum.inch);
 }
```

69.Add two complex numbers

```
#include<stdio.h>
typedef struct complex
{ float real;
float imag; }complex;
complex add(complex n1,complex n2);
void main()
{ complex n1,n2,temp;
 printf("For 1st complex number \n");
 printf("Enter real and imaginary respectively:\n");
scanf("%f%f",&n1.real,&n1.imag);
printf("\nFor 2nd complex number \n");
printf("Enter real and imaginary respectively:\n");
scanf("%f%f",&n2.real,&n2.imag);
temp=add(n1,n2);
printf("Sum=%.1f+%.1fi",temp.real,temp.imag);
  }
complex add(complex n1,complex n2)
{ complex temp;
temp.real=n1.real+n2.real;
temp.imag=n1.imag+n2.imag;
return(temp); }
```

70.Calculate difference between two time periods

```
#include <stdio.h>
struct TIME
{ int seconds;
```

```
 int minutes;
 int hours; };
void Difference(struct TIME t1, struct TIME t2, struct
TIME *diff);
 void main()
{ struct TIME t1,t2,diff;
 printf("Enter start time: \n");
printf("Enter hours, minutes and seconds respectively:
");
scanf("%d%d%d",&t1.hours,&t1.minutes,&t1.seconds)
; printf("Enter stop time: \n"); printf("Enter hours,
minutes and seconds respectively: ");
scanf("%d%d%d",&t2.hours,&t2.minutes,&t2.seconds)
; Difference(t1,t2,&diff);
 printf("\nTIME DIFFERENCE: %d:%d:%d -
",t1.hours,t1.minutes,t1.seconds);
printf("%d:%d:%d ",t2.hours,t2.minutes,t2.seconds);
printf("=
%d:%d:%d\n",diff.hours,diff.minutes,diff.seconds);  }
void Difference(struct TIME t1, struct TIME t2, struct
TIME *differ)
{ if(t2.seconds>t1.seconds)
{ --t1.minutes;
t1.seconds+=60; }
differ->seconds=t1.seconds-t2.seconds;
if(t2.minutes>t1.minutes)
{ --t1.hours;
t1.minutes+=60; }
 differ->minutes=t1.minutes-t2.minutes;
 differ->hours=t1.hours-t2.hours; }
```

## VI. Programs Using Strings

71.Program to Count Blanks,Tabs and Newlines

```
#include<stdio.h>
int main()
{       int nb,nt,nl;
  char c;
        nb=nt=nl=0;
        printf("\n Enter * to stop");
        while((c=getchar())!='*')
        {  if(c==' ')
                        ++nb;
                if(c=='\t')
                        ++nt;
                if(c=='\n')
                        ++nl;
  }
        printf("No. of Blanks is %d,No. of Tabs is %d
and No. of Newlines is %d",nb,nt,nl);
}
```

72.Palindrome checking

```
#include<stdio.h>
#include<string.h>
 void main()
{ int j,i,k,c=0;
 char a[80];
 printf("\nEnter main string:-\n");
 gets(a);
 k=strlen(a);
for(i=0,j=k-1;i<k/2;i++,j--)
{ if(a[i]==a[j])
   c++; }
  if(c==k/2)
  printf("Polyndrome");
  else
printf("\not Polyndrome");
}
```

73.convert a name into its ascii values.

```
#include<stdio.h>
 void main()
{char a[25];
int i=0;
printf("enter your name\n");
scanf("%s",a);
while(a[i]!='\0')
{printf("%c=%d\n",a[i],a[i]);
i++;}
}
```

74.calculating string length without strlen function

```
#include<stdio.h>
 void main()
{int i=1;
char a[25];
printf("Any Word\n");
while((a[i]=getchar())!='\n')
i++;
printf("lenght is %d",i-1);
}
```

75.comparing 2 strings without strcmp function

```
#include<stdio.h>
void main()
{
 int i,j,k=0,l,ls;
 char a[80],b[80];
 printf("\nEnter string1:-\n");
 gets(a);
 printf("\nEnter string2:-\n");
 gets(b);
 l=strlen(b);
ls=strlen(a);
 for(i=0,j=0;(i<l-1)||(j<ls-1);i++,j++)
  {
            if(a[i]==b[j])
             k=1;
            if(a[i]!=b[j])
              {k=0;
                      break;
                      }
 }
              if (k==1)
            printf("strings are equal\n");
            else
            {if(k==0)
            printf("\n\nstrings are not equal.");}

}
```

76 .copying one string to another without using strcpy

```
#include<stdio.h>
 void main()
{ int i,j,ls;
 char a[80],b[80];
 printf("\nEnter main string:-\n");
 gets(b);
 ls=strlen(b);
 for(i=0;i<=ls;i++)
  a[i]=b[i];
            printf("\n\ncopied string is %s ",a);
  }
```

77.string concatenation without using strcat function

```
#include<stdio.h>
 void main()
{ int i,j,l,ls;
 char a[80],b[80];
printf("\nEnter main string:-\n");
 gets(a);
 printf("enter the string to be concatinated\n");
 gets(b);
 l=strlen(a);
 ls=strlen(b);
 for(i=l,j=0;j<=ls;i++,j++)
  a[i]=b[j];
            printf("\n\nConcatinated string is ");
           puts(a);
}
```

78.Pattern replacement

```
#include<stdio.h>
 void main()
{char str[200],pat[20],new_str[200],rep_pat[100];
int i=0,j=0,k,n=0,rep=0;
printf("enter source string");
gets(str);
printf("enter string to be replaced ");
gets(pat);
printf("\n enter new string to replace pattern");
gets(rep_pat);
while(str[i]!='\0')
{
        j=0;k=i;rep=0;
        while(str[k]==pat[j] && pat[j]!='\0')
        {
        k++; j++;
        }
        if(pat[j]=='\0')
        {
        i=k;
                while(rep_pat[rep]!='\0')
                {
                new_str[n]=rep_pat[rep];
                rep++;
                n++;
                }
        }
new_str[n]=str[i];
i++;
n++;
}
new_str[n]='\0';
printf("The String is  ");
puts(new_str);
}
```

79.Counting the word occurance in a string

```
#include<stdio.h>
 #include<string.h>
main()
 {
  int strln,wordln,i,j,k,flag,count=0;
  char str[200],word[20];
  printf("Enter line of text:\n");
  gets(str);
  printf("Enter the word to count:\n");
  scanf("%s",word);
  strln=strlen(str);
  wordln=strlen(word);
  for(i=0;i<strln;i++)
  {
   if(str[i]==word[0]&&((str[i-1]=='
'||i==0)&&(str[i+wordln]==' '||str[i+wordln]=='\0')))
   {
    flag=0;k=i+1;
   for(j=1;j<wordln;j++,k++)
    {
     if(str[k]==word[j])
     {
      flag++;
     }
    }
    if(flag==wordln-1)
    {
     count++;
    }
   }
  }
  printf("Number of occurence of '%s' =
%d\n",word,count);
 }
```

79.Finding consecutive vowels

```
#include<stdio.h>
 void main()
{ int n,i,f=0,k=0;
 char a[80];

 printf("\nEnter main string:-\n");
 gets(a);
 n=strlen(a);
for(i=0;i<n;i++)
{
```

```
 if(a[i]=='a'||a[i]=='e'||a[i]=='i'||a[i]=='o'||a[i]=='u')
 {
  k=1;
if(a[i+1]=='a'||a[i+1]=='e'||a[i+1]=='i'||a[i+1]=='o'||a[i+1]
=='u')
  {printf("vowals %c and %c are found in position
%d\n",a[i],a[i+1],i+1);
   f=1;}
  }
  }
if(f==0)
{ if(k==1)
  printf("vowals found seperately\n");
  printf("\n vowals are not found consequitively");}
}
```

80.Sorting in alphabetical order

```
#include <stdio.h>
#include <string.h>
int main()
{ int i,j,n;
char a[10][20],t[20];
printf("Enter the number of strings :");
scanf("%d",&n);
for(i=0;i<n;i++)
scanf("%s",a[i]);// read the strings
for(i=0;i<n-1;i++) //bubble sort
for(j=0;j<n-1-i;j++)
if(strcmp(a[j],a[j+1])>0)
{ strcpy(t,a[j]);
strcpy(a[j],a[j+1]);
strcpy(a[j+1],t);
}
printf("The strings after sorting are :  \n");
for(i=0;i<n;i++)
{printf(" %s ",a[i]);// print the strings
printf("\n");}
}
```

81.Searching sub string in a string

```
#include<stdio.h>
void main()
{       char str[80],search[10];
        int count1=0,count2=0,i,j,flag;
        puts("Enter a string:");
        while ((str[count1]=getchar())!='\n')
                count1++;
        puts("Enter search substring:");
        while ((search[count2]=getchar())!='\n')
                count2++;
        for(i=0;i<=count1-count2;i++)
        {for(j=i;j<i+count2;j++)
                {flag=1;
                  if (str[j]!=search[j-i])
                   {flag=0;
                        break;    }
                }
                if (flag==1)
                        break;    }
        if (flag==1)
                puts("SEARCH SUCCESSFUL!");
        else
                puts("SEARCH
UNSUCCESSFUL!");
}
```

82. Find the frequency of a character in a string

```
#include <stdio.h>
void main()
{ char c[1000],ch;
 int i,count=0;
printf("Enter a string: ");
gets(c);
printf("Enter a characeter to find frequency: ");
scanf("%c",&ch);
for(i=0;c[i]!='\0';++i)
{ if(ch==c[i])
++count; }
printf("Frequency of %c = %d", ch, count);  }
```

83.Remove character in string, except alphabets

```
#include <stdio.h>
#include<stdio.h>
void main()
{ char line[150];
int i,j;
printf("Enter a string: ");
gets(line);
for(i=0; line[i]!='\0'; ++i)
{ while (!((line[i]>='a'&&line[i]<='z') ||
(line[i]>='A'&&line[i]<='Z' || line[i]=='\0')))
 { for(j=i;line[j]!='\0';++j)
{ line[j]=line[j+1]; }
 line[j]='\0'; }
 }
 printf("Output String: ");
 puts(line);
  }
```

84.Reverse the string

```
#include<stdio.h>
void main(){
```

```
char str[50];
char rev[50];
int i=-1,j=0;
printf("Enter any string : ");
scanf("%s",str);
   while(str[++i]!='\0');
while(i!=0)
 rev[j++] = str[--i];
rev[j]='\0';
  printf("Reverse of string is : %s",rev);
 }
```

## VII. Programs Using Pointers

**85.Area and perimeter of circle using pointers**

```
#include<stdio.h>
void areaperi ( int r, float *a, float *p )
{*a = 3.14 * r * r ;
*p = 2 * 3.14 * r ;
}
void main( )
{int radius ;
float area, perimeter ;
printf ( "\nEnter radius of a circle " ) ;
scanf ( "%d", &radius ) ;
areaperi ( radius, &area, &perimeter ) ;
printf ( "\nArea = %f", area ) ;
printf ( "\nPerimeter = %f", perimeter ) ;
}
```

**86.To check whether a number is prime using function pointers**

```
#include<stdio.h>
void isprime(int);
void (*fprime)(int);
void main()
{int n,i,j;
fprime=isprime;
printf("\n Enter the number ");
scanf("%d",&n);
(*fprime)(n);
getchar();
}
void isprime(int a)
{int i,fg=0;
for(i=2;i<a;i++)
{if(a%i==0)
fg=1;}
if (fg==0)
printf("\n prime");
else
printf("\n not prime");
}
```

**87.Duplication Removal using pointer**

```
#include<stdio.h>
void main(){
  int arr[50];
  int *p;
  int i,j,k,size,t;
  printf("\nEnter size of the array: ");
  scanf("%d",& size);
  printf("\nEnter %d elements into the array: ",size);
  for(i=0;i< size;i++)
    scanf("%d",&arr[i]);
  p=arr;
  for(i=0;i<size;i++){
    for(j=0;j<size;j++){
        if(i==j){
           continue;
        }
        else if(*(p+i)==*(p+j)){
           k=j;
           size--;
           while(k < size){
              *(p+k)=*(p+k+1);
              k++;
            }
           j=0;
         }      }  }
for(i=0;i<size;i++)
{            for(j=i+1;j<size;j++)
{   if(*(p+i)>*(p+j))
             {   t=*(p+i);
             *(p+i)=*(p+j);
             *(p+j)=t;}    }
}
  printf("\nThe array after removing duplicates is: ");
  for(i=0;i < size;i++)
    printf(" %d  ",arr[i]);
}
```

**88. Sorting integer arrays using pointers**

```
#include<stdio.h>
void sort(int size,int *p);
void main()
{
clrscr();
int i,a[8]={11,2,34,57,890,44,33,22};
sort(8,a);
for(i=0;i<8;i++)
printf("\n%d",a[i]);
}
void sort(int size,int *p)
{int j,t,i;
for(i=0;i<size;i++)
{for(j=i+1;j<size;j++)
{if(*(p+i)>*(p+j))
{t=*(p+i);
*(p+i)=*(p+j);
*(p+j)=t;
}}}}
```

**89.Sum of array using pointers**

```
#include<stdio.h>
 void main()
{ int a[10],i,sum=0;
clrscr();
 int *ptr;
 printf("Enter 10 elements:");
 for(i=0;i<10;i++)
 scanf("%d",&a[i]);
 ptr = a;          /* a=&a[0] */
```

```
 for(i=0;i<10;i++)
   {    sum = sum + *ptr;    //*p=content pointed by 'ptr'
   ptr++;    }
 printf("The sum of array elements is %d",sum);
}
```

**90.Count number of space,words,digits,numbers using pointers**

```
#include<stdio.h>
#include<stdlib.h>
#include<ctype.h>
 /*low implies that position of pointer is within a
word*/
#define low 1
/*high implies that position of pointer is out of word.*/
#define high 0
void main()
{
int nob,now,nod,nov,nos,pos=high;
char *s;
nob=now=nod=nov=nos=0;
printf("Enter any string:");
gets(s);
while(*s!='\0')
{
if(*s==' ')  /* counting number of blank spaces. */
   {    pos=high;
   ++nob;    }
else if(pos==high) /* counting number of words. */
   {    pos=low;
   ++now;    }
if(isdigit(*s))   /* counting number of digits. */
   ++nod;
if(isalpha(*s))    /* counting number of vowels */
   switch(*s)
            { case 'a':
            case 'e':
            case 'i':
            case 'o':
            case 'u':
            case 'A':
            case 'E':
            case 'I':
            case 'O':
            case 'U':
            ++nov;
            break;       }
/* counting number of special characters */
if(!isdigit(*s)&&!isalpha(*s))
   ++nos;
s++;}
printf("\nNumber of words %d",now);
printf("\nNumber of spaces %d",nob);
printf("\nNumber of vowels %d",nov);
printf("\nNumber of digits %d",nod);
printf("\nNumber of special characters %d",nos);
}
```

**91.Length of a string using pointer**

```
#include<stdio.h>
 int string_ln(char*);
void main()
{ char str[20];
  int l;
  printf("Enter any string: ");
  gets(str);
  l=string_ln(str);
  printf("The length of the given string %s is :
%d",str,l);
}
int string_ln(char*p)  /* p=&str[0] */
{  int count=0;
  while(*p!='\0')
  {    count++;
    p++;  }
  return count;
}
```

**92.Reverse the String Using Pointers**

```
#include<stdio.h>
void main()
{    char str[50], rev[50];
    char *sptr = str, *rptr = rev;
    int i=-1;
    printf("Enter any word : ");
    scanf("%s",str);
     while(*sptr){
     sptr++;
     i++;    }
    while(i>=0)
{   sptr--;
     *rptr = *sptr;
     rptr++;
     --i;    }
    *rptr='\0';
      printf("Reverse of string is : %s",rev);
    }
```

## VIII. Miscellaneous

93. Dec to Bin using bits

```
#include<stdio.h>
 void binary(unsigned int);   // Prototype Declaration
void main()
{unsigned int num;
printf("Enter Decimal Number : ");
scanf("%u",&num);
binary(num);   // Function Call
}
void binary(unsigned int num)
{unsigned int mask=32768;   //mask = [1000 0000 0000 0000]
printf("Binary Eqivalent : ");
while(mask > 0)
  {   if((num & mask) == 0 )
       printf("0");
  else
       printf("1");
 mask = mask >> 1 ;
  }
}
```

**94.Find Largest element element using dynamic memory allocation**

```
#include <stdio.h>
#include<stdlib.h>
void main()
{ int i,n;
float *data;
printf("Enter total number of elements(1 to 100): ");
scanf("%d",&n);
data=(float*)calloc(n,sizeof(float));
 /* Allocates the memory for 'n' elements */
if(data==NULL)
 { printf("Error!!! memory not allocated.");
exit(0); }
printf("\n");
for(i=0;i<n;i++)
{ printf("enter no: %d",i+1);
scanf("%f",data+i); }
for(i=0;i<n;i++)
{if(*data<*(data+i))
*data=*(data+i);
}
printf("Largest element = %.2f",*data);  }
```

95. Matrix multiplication using dynamic memory allocation

```
#include <stdio.h>
#include<stdlib.h>
void main()
{/* Declaring pointer for matrix multiplication.*/
int **ptr1, **ptr2, **ptr3;
/* Declaring integer variables for row and columns of two matrices.*/
int row1, col1, row2, col2;
/* Declaring indexes. */
int i, j, k;
/* Request the user to input number of columns of the matrices.*/
printf("\nEnter number of rows for first matrix : ");
scanf("%d", &row1);
printf("\nEnter number of columns for first matrix : ");
scanf("%d", &col1);
printf("\nEnter number of rows for second matrix : ");
scanf("%d", &row2);
printf("\nEnter number of columns for second matrix : ");
scanf("%d", &col2);
if(col1 != row2)
{printf("\nCannot multiply two matrices.");
return(0);
}
/* Allocating memory for three matrix rows. */
ptr1 = (int **) malloc(sizeof(int *) * row1);
ptr2 = (int **) malloc(sizeof(int *) * row2);
ptr3 = (int **) malloc(sizeof(int *) * row1);
/* Allocating memeory for the col of three matrices. */
for(i=0; i<row1; i++)
ptr1[i] = (int *)malloc(sizeof(int) * col1);
for(i=0; i<row2; i++)
ptr2[i] = (int *)malloc(sizeof(int) * col2);
for(i=0; i<row1; i++)
ptr3[i] = (int *)malloc(sizeof(int) * col2);
/* Request the user to input members of first matrix. */
printf("\nEnter elements of first matrix :\n");
for(i=0; i< row1; i++)
{for(j=0; j< col1; j++)
{printf("\tA[%d][%d] = ",i, j);
scanf("%d", &ptr1[i][j]);}}
/* request to user to input mebmbers of second matrix. */
printf("\nEnter elements of second matrix :\n");
for(i=0; i< row2; i++)
{for(j=0; j< col2; j++)
{printf("\tB[%d][%d] = ",i, j);
scanf("%d", &ptr2[i][j]);}}
/* Calculation begins for the resultant matrix. */
for(i=0; i < row1; i++)
```

```
{for(j=0; j < col1; j++)
{ptr3[i][j] = 0;
for(k=0; k<col2; k++)
ptr3[i][j] = ptr3[i][j] + ptr1[i][k] * ptr2[k][j];
}
}
/* Printing the contents of third matrix. */
printf("\n\nResultant matrix :");
for(i=0; i< row1; i++)
{printf("\n\t\t\t");
for(j=0; j < col2; j++)
   printf("%4d", ptr3[i][j]);}
return(0);}
```

97.Reverse the digit without using % operator

```
#include<stdio.h>
 #include<stdlib.h>
#include<string.h>
void main()
{ int num1, num2;
 char str[10];
     printf("\nEnter the Number : ");
    scanf("%d",&num1);
    sprintf(str,"%d",num1);
    strrev(str);
    num2 = atoi(str);
    printf("\nReversed Number : ");
    printf("%d",num2);
}
```

96. Add Digits of the Number Using Single Statement :

```
#include<stdio.h>
void main()
{int number=12354;
int sum=0;
for(;number > 0;sum+=number%10,number/=10);
printf("\nSum of the Digits : %d",sum);
}
```

**98.Addition without using +**

```
#include<stdio.h>
 void main()
{int a=10,b=5;
a = a-(-b);
printf("Sum is : %d", a);
}
```

**99.Addition without using arithmetic operators**

```
#include<stdio.h>
 void main()
{int a=10,b=5;
while(b--)
   a++;
printf("Sum is : %d", a);
}
```

**100. Stack operation**

```
#include<stdio.h>
 #define max 10
int st[max],top=-1;
void push(int st[],int val);
int pop(int st[]);
int peep(int st[]);
void display(int st[]);
void main()
{int val,opt;
do
{printf("\n 1.push \n 2.pop \n3.peep\n 4.display\n
5.exit");
scanf("%d",&opt);
switch(opt)
{case 1:
printf("enter value to be pushed\n");
```

```
scanf("%d",&val);
push(st,val);
break;
case 2:
val=pop(st);
printf("the value deleted from stack is %d", val);
break;
case 3:val=peep(st);
printf("the value stored in top of stack is %d", val);
break;
case 4:
display(st);
break;
}}while(opt<5);
}
void push(int st[], int val)
{if(top==max-1)
printf("overflow");
else
{top++;
st[top]=val;
}}
int pop(int st[])
{int val;
if (top==-1)
{printf("stank underflow");
return (-1);}
else
{val=st[top];
top--;
return val;}
}
void display(int st[])
{int i;
if(top==-1)
printf("stack is empty");
else
{for(i=top;i>=0;i--)
printf("\n%d",st[i]);}
}
int peep(int st[])
{if (top== -1)
{printf("stack is empty");
return (-1);}
else return(st[top]);
}
```

**101.Printing Patterns**

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

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

**102**

```
#include <stdio.h>
int main()
{
   int i, j, rows;
   printf("Enter number of rows: ");
   scanf("%d",&rows);
   for(i=1; i<=rows; ++i)
   {
      for(j=1; j<=i; ++j)
      {
         printf("%d ",j);
      }
      printf("\n");
   }
   return 0;
}
```

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

**103.**

```
#include <stdio.h>
int main()
{
   int i, j;
   char input, alphabet = 'A';
   printf("Enter the uppercase character you want to
print in last row: ");
   scanf("%c",&input);
   for(i=1; i <= (input-'A'+1); ++i)
   {
      for(j=1;j<=i;++j)
            printf("%c", alphabet);
      ++alphabet;
      printf("\n");
   }
}
```

```
A
B B
```

```
C C C
D D D D
E E E E E
```

```
104.#include <stdio.h>
int main()
{
   int i, j, rows;
   printf("Enter number of rows: ");
   scanf("%d",&rows);
   for(i=rows; i>=1; --i)
   {
      for(j=1; j<=i; ++j)
      {
         printf("* ");
      }
      printf("\n");
   }
}
```

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

```
105.#include <stdio.h>
int main()
{
   int i, j, rows;
   printf("Enter number of rows: ");
   scanf("%d",&rows);

   for(i=rows; i>=1; --i)
   {
      for(j=1; j<=i; ++j)
      {
         printf("%d ",j);
      }
      printf("\n");
   }
}
```

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

```
106.#include <stdio.h>
int main()
{
   int i, space, rows, k=0;
   printf("Enter number of rows: ");
   scanf("%d",&rows);

   for(i=1; i<=rows; ++i, k=0)
   {
      for(space=1; space<=rows-i; ++space)
      {
         printf("  ");
      }
      while(k != 2*i-1)
      {
         printf("* ");
         ++k;
      }
      printf("\n");
   }
}
```

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

```
107.#include <stdio.h>
int main()
{
   int i, space, rows, k=0, count = 0, count1 = 0;
   printf("Enter number of rows: ");
   scanf("%d",&rows);
   for(i=1; i<=rows; ++i)
   {
      for(space=1; space <= rows-i; ++space)
      {
         printf("  ");
         ++count;
      }
      while(k != 2*i-1)
      {
         if (count <= rows-1)
         {
            printf("%d ", i+k);
            ++count;
         }
         else
         {
            ++count1;
            printf("%d ", (i+k-2*count1));
         }
         ++k;
      }
```

```
        count1 = count = k = 0;
        printf("\n");
    }
}
```

```
      1
    2 3 2
  3 4 5 4 3
 4 5 6 7 6 5 4
5 6 7 8 9 8 7 6 5
```

108.

```
#include<stdio.h>
int main()
{
    int rows, i, j, space;
    printf("Enter number of rows: ");
    scanf("%d",&rows);
    for(i=rows; i>=1; --i)
    {
        for(space=0; space < rows-i; ++space)
            printf("  ");
        for(j=i; j <= 2*i-1; ++j)
            printf("* ");
        for(j=0; j < i-1; ++j)
            printf("* ");
        printf("\n");
    }
    return 0;
}
```

```
       1
      1   1
     1   2   1
    1   3   3           1
  1  4    6   4   1
 1  5   10   10  5   1
```

109.

```
#include <stdio.h>
int main()
{
    int rows, i, j, number= 1;
    printf("Enter number of rows: ");
    scanf("%d",&rows);
    for(i=1; i <= rows; i++)
    {
        for(j=1; j <= i; ++j)
        {
            printf("%d ", number);
            ++number;
        }
        printf("\n");
    }
    return 0;
}
```

```
1
2 3
4 5 6
7 8 9 10
```

110.Sum of 2 matrix using dynamic memory allocation

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

int main()
{
    int i,*ptr1[3],*ptr2[3],*ptr3[3],j;
    for(i=0;i<3;i++)//dynamic init
    {ptr1[i]=(int*)malloc(3*sizeof(int));
     ptr2[i]=(int*)malloc(3*sizeof(int));
     ptr3[i]=(int*)malloc(3*sizeof(int));}
     printf("\n\Enter mat 1\n");
    for(i=0;i<3;i++)
        for(j=0;j<3;j++)
        scanf("%d",(*(ptr1+i)+j));
    printf("\n\Enter mat 2\n");
    for(i=0;i<3;i++)
        for(j=0;j<3;j++)
        scanf("%d",(*(ptr2+i)+j));
        printf("\n\Enter sum is: \n");
    for(i=0;i<3;i++)
    {
         for(j=0;j<3;j++)
         *(*(ptr3+i)+j)=*(*(ptr1+i)+j)+*(*(ptr2+i)+j);
         printf("\n");
    }
    for(i=0;i<3;i++)
    {
         for(j=0;j<3;j++)
         printf("%d ",*(*(ptr3+i)+j));
         printf("\n");
    }
    free(ptr1);
    free(ptr2);
    free(ptr3);

}
```

111.Employee structure sorting based on salary

```
#include<stdio.h>
struct employee
{
    char name[200];
    int salary;
};
int main()
```

```
{
   int n,i,j;
   struct employee e[20],t;
   printf("Enter N:");
   scanf("%d",&n);
   for(i=0;i<n;i++)
   {
      printf("Enter the name and salary for employee %d
\n",i+1);
      scanf("%s%d",e[i].name,&e[i].salary);
   }
   for(i=0;i<n-1;i++)
   {
       for(j=0;j<n-i-1;j++)
       {

          if(e[j].salary>e[j+1].salary)
          {
            t=e[j];
            e[j]=e[j+1];
            e[j+1]=t;
          }
       }
   }
   for(i=0;i<n;i++)
   {
      printf("\nname:%s \t
salary:%d",e[i].name,e[i].salary);
   }

}
```

**112.Merge sorting using dynamic memory allocation**

```
#include <stdio.h>
#include<stdlib.h>
 int main()
   {

      int *array1, *array2, *array3, m, n, i, j, k = 0;
      printf("\n Enter size of array Array 1: ");
      scanf("%d", &m);
      array1=(int*)calloc(m,sizeof(int));
      printf("\n Enter sorted elements of array 1: \n");
      for (i = 0; i < m; i++)
      {
         scanf("%d", array1+i);
      }

      printf("\n Enter size of array 2: ");
      scanf("%d", &n);
       array2=(int*)calloc(n,sizeof(int));
      printf("\n Enter sorted elements of array 2: \n");
      for (i = 0; i < n; i++)
      {
         scanf("%d", array2+i);
      }

      i = 0;
      j = 0;

      while (i < m && j < n)
      {
         if (array1[i] < array2[j])
         {
            array3[k] = array1[i];
            i++;
         }

         else
         {
            array3[k] = array2[j];
            j++;
         }
         k++;
      }

      if (i >= m)
      {
         while (j < n)
         {
            array3[k] = array2[j];
            j++;
            k++;
         }
      }

      if (j >= n)
      {
         while (i < m)
         {
            array3[k] = array1[i];
            i++;
            k++;
         }
      }

      printf("\n After merging: \n");
      for (i = 0; i < m + n; i++)
      {
         printf("\n%d", array3[i]);
      }
   return 0;
     }
```

113.**Addition of quadratic equations using structures**

```
#include<stdio.h>
struct ply
{
        int ex,co;
}p[3];
int main()
{
        int i;
        struct ply p1[3],p2[3];
        printf("Enter the exponent and coefficient for
polynomial 1");
        for(i=0;i<3;++i)
        {       scanf("%d %d",&p1[i].ex,&p1[i].co);
        }
        printf("Enter the exponent and coefficient for
polynomial 2");
        for(i=0;i<3;++i)
        {       scanf("%d %d",&p2[i].ex,&p2[i].co);
        }
        for(i=0;i<3;++i)
        {
                p[i].co=p1[i].co+p2[i].co;
                p[i].ex=((p1[i].ex+p2[i].ex)/2);
        }
        printf("\n Sum of the polynomial is");
                for(i=0;i<3;++i)
        {       printf("\n %d %d \n",p[i].ex,p[i].co);
        }
}
```

**104.Selection Sort**

```
#include <stdio.h>
int main()
{
 int array[100], n, c, d, position, swap;
 printf("Enter number of elements\n");
 scanf("%d", &n);
 printf("Enter %d integers\n", n);
 for (c = 0; c < n; c++)
  scanf("%d", &array[c]);
 for (c = 0; c < (n - 1); c++)
 {
  position = c;
  for (d = c + 1; d < n; d++)
  {
   if (array[position] > array[d])
    position = d;
  }
  if (position != c)
  {
   swap = array[c];
   array[c] = array[position];
   array[position] = swap;
  }
 }

 printf("Sorted list in ascending order:\n");
 for (c = 0; c < n; c++)
  printf("%d\n", array[c]);
 return 0;
}
```