

SASTRA University

C-Programs

I-Btech

I. Simple Programs	--03	46. Matrix multiplication	
1. Find area of a rectangle		47. Inverse of a 3X3 matrix	
2. Find ASCII value of a character		IV. Programs using Functions	--10
3. Convert Celsius to Fahrenheit		48. Factorial using function	
4. Swap value of two variables		49. Min and Max of array	
5. Check the given number is odd or even		50. Bubble Sort	
6. Check whether a character is vowel or consonant		51. Convert :Bin to dec; dec to bin	
7. Find largest among three numbers		52. Bin to oct; oct to bin	
8. Leap year checking		53. Dec to Hex	
9. Positive negative checking		54. Oct to dec; dec to oct	
10. Quadratic equation		55. Stack operation using function	
II. Programs using Loops	--04	56. Factorial using recursive function	
11. Factorial without using function		57. Fibonacci using recursive function	
12. Table of N and square of N		58. Sum of N numbers using recursion	
13. Calculate x to the power y		59. Reverse the sentence using recursion	
14. Multiplication table		60. Power using recursion	
15. Sum of natural numbers		61. Towers of Hanoi	
16. Fibonacci starting from any two numbers		62. Exponent using recursion	
17. Upper case to Lower case		63. GCD using recursion	
18. Lower to upper		V. Programs using Structures	--14
19. Pascal triangle		64. Student structure	
20. LCM & GCD		65. Players structure	
21. Prime numbers between two ranges		66. Add two polynomials using structures in function	
22. Factors of a number		67. Add two distances using structures	
23. Prime Factors		68. Add two complex numbers	
24. Bin to dec and oct		69. Calculate difference between two time period	
25. Count the number of digit in an integer		VI. Programs using Strings	--16
26. Reverse the digits of given number		70. Program to Count Blanks, Tabs and Newlines	
27. Number palindrome		71. Palindrome checking	
28. Digit summation		72. convert a name into its ascii values.	
29. Armstrong checking		73. calculating string length without strlen function	
30. Make simple calculator in C		74. comparing 2 strings without strcmp function	
31. TO FIND SIN(X) USING SINE SERIES		75. copying one string to another without using strcpy	
32. Exponent series		76. string concatenation without using strcat function	
33. Floyd's Triangle		77. Pattern replacement	
III. Programs using Arrays	--07	78. Finding vowels	
34. Fibonacci using array		79. Sorting in alphabetical order	
35. Largest among N numbers in an array		80. Searching sub string in a string	
36. Smallest among N numbers in an array		81. Find the frequency of a character in a string	
37. Reverse the array elements		82. Remove characters in string except alphabets	
38. Insert an element in an array		83. Reverse the given string	
39. Deleting an array element		VII. Programs using Pointers	--19
40. Transpose of a matrix		84. Area of circle using pointers	
41. Duplication removal		85. function pointers	
42. Linear Search		86. duplication removal using pointers	
43. Binary search		87. Sorting integer array using pointers	
44. Split the sorted array		88. Sum of array using pointers	
45. Matrix addition			

- 89. Count number of words using pointers
- 90. Length of a string using pointers
- 91. Reverse the String Using Pointers
- VIII. Programs using Files --21**
- 92. Write a sentence into a file
- 93. Employee file
- 94. Employee-struct
- 95. Copying the content of one file into another
- 96. Convert the file contents in Upper-case & Write Contents in a output file
- 97. Compare two text/data files in C Programming
- 98. Reading & writing in files
- 99. ODD-EVEN splitting
- 100. Copy from one text file into another

- 101. Display same source code as output
- 102. Read a string of text from a file
- IX. Miscellaneous -30**
- 103. Multiple files-Prime factors
- 104. Multiple files-String sort & search
- 105. Dec to Bin using bits
- 106. Find Largest element using dynamic memory allocation
- 107. Matrix multiplication using dynamic memory allocation
- 108. Add Digits of the Number Using Single Statement
- 109. Reverse the digit without using % operator
- 110. Addition without using +
- 111. Addition without using arithmetic operators

I. Simple Programs

1. Find area of rectangle

```
#include<stdio.h>
void main()
{ int length,breadth,side;
clrscr(); // Clear Screen
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>
void 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>
void 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>
void 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>
void 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>
void 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>
void 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>=a && b>=c)
printf("Largest number = %.2f", b);
else printf("Largest number = %.2f", c);
}
```
8. Leap year checking

```
#include<stdio.h>
void 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. Positiv negative checking

```
#include <stdio.h>
void 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;
clrscr();
printf("value 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>
void 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>
void main()
{ int n;
printf("\nnot Squaren");
printf("-----n");
for(n=1;n <=10;n++)
    printf("%d\t%d\n",n,n*n);
}
```

13. Calculate x to the power y

```
#include <stdio.h>
void 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>
void 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>
void 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>
void 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>
#include<string.h>
void main()
{ char str[20];
int i;
printf("Enter any string->");
scanf("%s",str);
printf("The string is->%s",str);
for(i=0;i<=strlen(str);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>
void main()
{ char str[20];
  int i;
  printf("Enter any string->");
  scanf("%s",str);
  printf("The string is->%s",str);
  for(i=0;i<=strlen(str);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>
void main()
{int bin=1,p,q=0,r,x;
printf("Rows you want to input:");
scanf("%d",&r);
printf("\nPascal'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>
# include <conio.h>
void 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>
void main()
{ int n1, n2, i, j, flag;
printf("Enter two numbers(intervals): ");
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)
Printf("%d ",i);}
}
```

22. Factors of a number

```
#include <stdio.h>
void 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>
void 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. Bin to dec and oct

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

```

26. Reverse the digits of given number

```

#include<stdio.h>
void 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>
void 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>
void 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*10+m; }
printf("the value is %d",sum);
}

```

29. Armstrong checking

```

#include<stdio.h>
void 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 armstrong number",ams);
else
printf("%ld is not a armstrong number ",ams);
}

```

30. Making simple calculator in C

```

#include <stdio.h>
void 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("\num1+num2=%f",num1+num2); break;
case '-': printf("\num1-num2=%f",num1-num2); break;
case '*': printf("\num1*num2=%f",num1*num2); break;
case '/': printf("\num2/num1 = %f",num1/num2); break;
default: printf("Error! operator is not correct"); break; }
}

```

31. Find sin(x) using series

```

#include<stdio.h>
#include<math.h>
void 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++)
{sum+=term;
term=-term*x*x/((2*i-1)*(2*i-2));}
printf("Sin (%f)=%f",xd,sum);
}

```

32. Exponent series

```

#include<stdio.h>
#define ACCURACY 0.0001
void 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>
void 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>
void 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>
void 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>
void 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>
void 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>
void 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]);
  getch();
  }

40. Transpose of a matrix
#include<stdio.h>
void 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("\n Given 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; }
  printf("\n Transpose 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>
void main()
{ int a[50], int i,j,k,size,n,t;
  printf("\n Enter size of the array: ");
  scanf("%d",&n);
  printf("\n Enter %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("\n The array after removing duplicates is: ");
  for(i=0;i < size;i++)
      printf(" %d ",a[i]);
  }

42. Linear Search
#include<stdio.h>
void main()
{ int a[30],x,n,i;
  printf("\n Enter no of elements :");
  scanf("%d",&n);
  printf("\n Enter the values :");
  for(i=0;i < n;i++)
      scanf("%d",&a[i]);
  printf("\n Enter 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>
void 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>
void 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>
void 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>
void main()
{int a[10][10],b[10][10],c[10][10],i,j,k;
int sum=0;
printf("\nEnter First Matrix : \n");
for(i=0;i<3;i++)
for(j=0;j<3;j++)
scanf("%d",&a[i][j]);
printf("\nEnter Second Matrix:\n");
for(i=0;i<3;i++)
for(j=0;j<3;j++)
```

```

scanf("%d",&b[i][j]);
for(i=0;i<=2;i++)
for(j=0;j<=2;j++)
{ sum = 0;
for(k=0;k<=2;k++)
sum = sum + a[i][k] * b[k][j];
c[i][j]=sum; }
printf("\nMultiplication Of Two Matrices : \n");
for(i=0;i<3;i++)
{ for(j=0;j<3;j++)
printf(" %d ",c[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; }
}
void 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 Functions

48. 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;
}

```

49. 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 (void)
{int array1[5] = { 157, -28, -37, 26, 10 };
int array2[7] = { 12, 45, 1, 10, 5, 3, 22 };
int minimum (int values[], int numberOfElements);
printf ("array1 minimum: %i\n", minimum (array1, 5));
printf ("array2 minimum: %i\n", minimum (array2, 7));
}

```

50. Bubble Sort

```

#include<stdio.h>
void bubble_sort(int [],int);
void 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);
getch();
}
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]); }
}

```

51. Convert :Bin to dec; dec to bin

```

#include <stdio.h>
#include <math.h>
int binary_decimal(int n);
int decimal_binary(int n);
void 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;
}

```

52. Bin to oct; oct to bin

```

#include <stdio.h>
#include <math.h>
int binary_octal(int n);
int octal_binary(int n);
void 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; }

```

53. Dec to Hex

```

#include<stdio.h>
#include<math.h>
void dec_hex(long int num) // Function Definition
{long int 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]);  }
    }}
void main()
{
    long int num;
    printf("Enter the decimal number : ");
    scanf("%ld",&num);
    dec_hex(num);
}

```

54. Oct to dec; dec to oct

```

#include <stdio.h>
#include<math.h>
int decimal_octal(int n);
int octal_deciaml(int n);
void 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; }

```

55. 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]);
}

```

56. Factorial using recursive function

```

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

```

58. Sum of N numbers using recursion

```

#include<stdio.h>
int add(int n);
void 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);
}

```

59. Reverse the sentence using recursion

```

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

```

60. Power using recursion

```

#include<stdio.h>
int power(int n1,int n2);
void 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;}

```

61. Towers 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);
}
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');
}

```

```

}

Towers of Hanoi- 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("Move diask from %c to %c",a,c);
tower(n-1,b,a,c);
}
}
void main()
{int n,I,step=1;
printf("enter number of disk");
scanf("%d",&n);
tower(n,'A','B','C');
for(i=1;i<=n;i++)
step=step*2;
printf("the number of steps used is %d",step-1);
}

```

62. Exponent using recursion

```

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

```

63. GCD using recursion

```

int GCD(int,int);
void main()
{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 Structures

64. Student details -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;
clrscr();
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].dob.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); }
}

```

65. Players detail- structure

```

#include<stdio.h>
struct play
{char name[25];
int age;
int nmatch;
int run;
float avgrun;
}cri[100];
void main()
{ int n,i;
float d;
clrscr();
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].nmatc
h,&cri[i].run); }
for(i=0;i<n;i++)
cri[i].avgrun=cri[i].run/cri[i].nmatch;
for(i=0;i<n;i++)

```

```

    { if(cri[i].avgrun>cri[i+1].avgrun)
      d=cri[i+1].avgrun;
      cri[i+1].avgrun=cri[i].avgrun;
      cri[i].avgrun=d; }
printf("\ndetails in ascending order\n");
printf("\nName\tage\tmatches\ttruns\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);
}

```

66. 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 add 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;
clrscr();
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);
}

```

67. 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 information 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);
}

```

68. 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); }
```

69. 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

70. Program to Count Blanks,Tabs and Newlines

```
#include<stdio.h>
int main(void)
{ int nb,nt,nl,c;
  nb=nt=nl=0;
  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);
}
```

71. Palindrome checking

```
#include<stdio.h>
void main()
{ int j,i,k,c=0;
  char a[80];
  clrscr();
  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");
  getch();
}
```

72. convert a name into its ascii values.

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

73. calculating string length without strlen function

```
#include<stdio.h>
void main()
{ int i=1;
  char a[25];
  printf("any number\n");
  while((a[i]=getchar())!='\n')
    i++;}
```



```
printf("length is %d",i-1);
getch();
}
```

74. comparing 2 strings without strcmp function
#include<stdio.h>

```
void main()
{
    int i,j,k=0,l,ls;
    char a[80],b[80];
    clrscr();
    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.");}
    getch();
}
```

75. copying one string to another without using strcpy

```
#include<stdio.h>
void main()
{ int i,j,l,ls;
  char a[80],b[80];
  clrscr();
  printf("\nEnter main string:-\n");
  gets(b);
  ls=strlen(b);
  for(i=0,j=0;j<=ls-1;i++,j++)
  a[i]=b[j];
  printf("\n\ncopied string is %s ",a);
}
```

76. 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 concatenated\n");
gets(b);
l=strlen(a);
ls=strlen(b);
for(i=1,j=0;j<=ls;i++,j++)
a[i]=b[j];
printf("\n\nconcatinated string is ");
puts(a);
getch();
}
```

77. 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++;
      }
      printf("The String is ");
      puts(new_str);
      getch();
  }
```

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);
strlen=strlen(str);
wordln=strlen(word);
for(i=0;i<strlen;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 occurrence of '%s' =
%d\\n",word,count);
}

```

78. Finding vowels

```

#include<stdio.h>
void main()
{ int n,i,f=0,k=0;
char a[80];
clrscr();
printf("\\n\\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("vowels %c and %c are found in position
%d",a[i],a[i+1],i+1);
        f=1;}
    }
}
if(f==0)
{ if(k==1)
    printf("vowels found separately");
    printf("\\n vowels are not found consequitively");}
}

```

79. Sorting in alphabetical order

```

#include <stdio.h>
#include <string.h>
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");}
}

```

80. Searching sub string in a string

```

#include<stdio.h>
void main()
{
    char str[80],search[10];
    int count1=0,count2=0,i,j,flag;
    clrscr();
    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!");
}

```

81. 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 character 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); }

```

82. Remove character in string, except alphabets

```

#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);
}

```

83. 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

84. Area 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 ) ;
}

```

85. function pointers

```

#include<stdio.h>
void isprime(int);
void (*fprime)(int);
void main()
{int n,i,j,c=0,k=1;
fprime=isprime;
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("prime");
else
printf("\not prime");
}

```

86. duplication removal using pointers

```

#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: ",n);
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]);
}

```

87. Sorting integer array using pointers

```

#include<stdio.h>
void sort(int size,int *p);
void main()
{ 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;
      } }
}

```

88. Sum of array using pointers

```

#include<stdio.h>
void main()
{ int a[10], int i,sum=0;
  int *ptr;
  printf("Enter 10 elements: \n");
  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);
}

```

89. 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!="")
  {
    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 'T':
      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);
}

```

90. Length of a string using pointer

```

#include<stdio.h>
int string_ln(char*);
void main()
{ char str[20];
  int l;

```

```

printf("Enter any string: \n");
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;
}

```

91. 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 string : ");
  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. Programs using Files

92. Write a sentence into a file

```

#include <stdio.h>
#include<stdlib.h> /* For exit() function */
void main()
{ char c[1000];
  FILE *fptr;
  fptr=fopen("program.txt","w");
  if(fptr==NULL)
  { printf("Error!");
    exit(1); }
  printf("Enter a sentence:\n");
  gets(c);
  fprintf(fptr,"%s",c);
  fclose(fptr);
}

```

93. Files to maintain employee details

```

#include<stdio.h>
#include<stdlib.h>
void main()
{ FILE *emp,*empsal;
  int id,sal;
  float da,d,net;

```

```

char na[100],add[100];
printf("id,name,address, sal\n");
scanf("%d",&id);
flushall();
gets(na);
gets(add);
scanf("%d",&sal);
emp=fopen("Employee.txt","w");
fprintf(emp,"\nID=%d\nname:%s\naddress:%s\nBP:%d",id
,na,add,sal);
fclose(emp);
emp=fopen("Employee.txt","r");
empsal=fopen("Emp_sal.txt","w");
fscanf(emp,"\nID=%d\nname:%s\naddress:%s\nBP:%d",id
,na,add,sal);
da=sal*0.5;
d=sal*0.1;
net=sal+da+d;
fprintf(empsal,"\nID=%d\nname:%s\naddress:%s\nBP:%d\n
n%f\n%f\n%f\n",id,na,add,sal,da,d,net );
printf("\n-----\n");
printf("ID: %d",id);
printf("\n name:");
puts(na);
printf("\naddress:");
puts(add);
printf("BP:%d\n DA: %.2f\n
Detec: %.2f\nNET: %.2f\n",sal,da,d,net);
fclose(emp);
fclose(empsal);
}

```

94. Files to process structure of Employee

```

#include<stdio.h>
struct employee
{ int empid;
  char name[25];
  int age;
  long int sal;
};
void main()
{ struct employee e[30],emp[30];
  FILE *fp;
  int i,num,n;
  printf("Enter the no of employees:");
  scanf("%d",&n);
  /*Read the details of three employees */
  fp=fopen("efile","w");
  for(i=0;i<n;i++)
  { printf("Enter the empid,name,age and salary:\n");
    scanf("%d %s %d %ld",&e[i].empid, e[i].name,
    &e[i].age,&e[i].sal); }
  //write all the records in the file
  fwrite(&e,sizeof(struct employee),n,fp);
  fclose(fp);
  fp=fopen("efile","r");

```

```
//read all records from the file
fread(&emp,sizeof(struct employee),n,fp);
printf("Employee details(sal<4500 and age>35):");
for(i=0;i<n;i++)
{if(emp[i].sal<4500 && emp[i].age>35)
printf("\n\nEmployee id = %d\nName = %s\nage = %d\nsal= %ld\n",emp[i].empid,emp[i].name,emp[i].age,emp[i].sal);
}
fclose(fp);
getch();}
```

95. Copying the content of one file into another

```
#include<stdio.h>
#include<process.h>
void main()
{FILE *fp1,*fp2;
char a;
fp1=fopen("test.txt","r");
if(fp1==NULL)
{ puts("cannot open this file");
exit(1); }
fp2=fopen("test1.txt","w");
if(fp2==NULL)
{ puts("Not able to open this file");
fclose(fp1);
exit(1); }
do
{ a=fgetc(fp1);
fputc(a,fp2);
}while(a!=EOF);
fcloseall();
}
```

96. Convert the file contents in Upper-case & Write Contents in a output file

```
#include<stdio.h>
#include<process.h>
void main()
{FILE *fp1,*fp2;
char a;
fp1=fopen("test.txt","r");
if(fp1==NULL)
{ puts("cannot open this file");
exit(1); }
fp2=fopen("test1.txt","w");
if(fp2==NULL)
{ puts("Not able to open this file");
fclose(fp1);
exit(1); }
do
{ a=fgetc(fp1);
a=toupper(a);
fputc(a,fp2);
}while(a!=EOF);
fcloseall();
}
```

}

97. Compare two text/data files in C Programming

```
#include<stdio.h>
void main()
{ FILE *fp1, *fp2;
int ch1, ch2;
char fname1[40], fname2[40];
printf("Enter name of first file :");
gets(fname1);
printf("Enter name of second file:");
gets(fname2);
fp1 = fopen( fname1, "r" );
fp2 = fopen( fname2, "r" );
if ( fp1 == NULL )
{ printf("Cannot open %s for reading ", fname1 );
exit(1); }
else if (fp2 == NULL)
{ printf("Cannot open %s for reading ", fname2 );
exit(1); }
else
{ ch1 = getc( fp1 );
ch2 = getc( fp2 );
while( (ch1!=EOF) && (ch2!=EOF) && (ch1 == ch2))
{ ch1 = getc(fp1);
ch2 = getc(fp2);
}
if (ch1 == ch2)
printf("Files are identical \n");
else if (ch1 != ch2)
printf("Files are Not identical \n");
fclose ( fp1 );
fclose ( fp2 );
}
return(0);
}
```

98. Read a string of text from a file

```
#include<stdio.h>
#include <stdlib.h> /* For exit() function*/
void main()
{ char c[1000];
FILE *fptr;
if ((fptr=fopen("program.txt","r"))==NULL)
{ printf("Error! opening file");
exit(1); /* Program exits if file pointer returns NULL. */ }
fscanf(fptr,"%[^\\n]",c);
printf("Data from file:\\n%s",c);
fclose(fptr);
}
```

99. Reading & writing in files

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

```

{ int roll;
  char name[12];
  int percent;
}s = { 10,"SMJC",80};
void main()
{FILE *fp;
struct stud s1;
fp = fopen("ip.txt","w");
/* write struct s to file */
fwrite(&s, sizeof(s), 1,fp);
fclose(fp);
fp = fopen("ip.txt","r");
/* read struct s to file */
fread(&s1, sizeof(s1), 1,fp);
fclose(fp);
printf("\nRoll : %d",s1.roll);
printf("\nName : %s",s1.name);
printf("\nPercent : %d",s1.percent);
}

```

100. ODD-EVEN Splitting

```

#include<stdio.h>
#include<stdlib.h>
void main()
{ FILE *mfile,*odd,*even;
int n,a[100],i,b;
printf("\no of elements\n");
scanf("%d",&n);
for(i=0;i<n;i++)
scanf("%d",&a[i]);
mfile=fopen("mainfile.txt","w");
for(i=0;i<n;i++)
fprintf(mfile,"\n%d",a[i]);
fclose(mfile);
mfile=fopen("mainfile.txt","r");
odd=fopen("oddfile.txt","w");
even=fopen("evenfile.txt","w");
while(!feof(mfile))
{ fscanf(mfile,"%d",&b);
if(b%2==0)
fprintf(even,"\n%d",b);
else
fprintf(odd,"\n%d",b);
}
fclose(mfile);
fclose(odd);
fclose(even);
}

```

101. Copy from one text file into another

```

#include<stdio.h>
#include<stdlib.h>
void main()
{ FILE *fp1,*fp2;
char ch;
fp1 = fopen("Sample.txt","r");
fp2 = fopen("Output.txt","w");

```

```

while(1)
{ ch = fgetc(fp1);
if(ch==EOF)
break;
else
putc(ch,fp2);
}
printf("File copied succesfully!");
fclose(fp1);
fclose(fp2);
}

```

102. Display same source code as output

```

#include<stdio.h>
void main(){
FILE *fp;
char c;
fp = fopen(__FILE__,"r");
do{
c= getc(fp);
putchar(c); }
while(c!=EOF);
fclose(fp);
}

```

103. Multiple files-Prime factors

```

(primeA.c)
extern int f[100];
void factor(int n)
{
int i,j=1;
for(i=2;i<=n;i++)
{if(n%i==0)
{f[j++]=i;
n=n/i;
i--;
if(n==1) break;} }
}

(primeB.c)
extern int n;
int isprime(int x)
{int i,count=0;
for(i=2;i<=x/2;i++){
if(x%i==0){
count++;
break; } }
if(count==0 && x>1)
return 1;
else
return 0;
}

(primemain.c)
#include<stdio.h>
#include"primeA.c"

```

```

#include"primeB.c"
int f[100];
void main()
{int i,n;
printf("Enter the value of n:");
scanf("%d",&n);
factor(n);
printf("\nThe prime factors are:");
for(i=1;i<=n;i++)
if(isprime(f[i]))
printf(" %d",f[i]);
}

```

104. Multiple files-String sort & search

```

//stringA- TO GET ARRAY OF STRINGS//
#include<stdio.h>
extern char s[30][30];
extern int n;
void getstring()
{int i;
printf("\nEnter strings:\n");
flushall();
for(i=0;i<n;i++)
gets(s[i]);
}

//stringB TO SORT ARRAY OF STRINGS//
extern char s[30][30];
extern int n;
void sort()
{int i,j;
char temp[30];
for(i=0;i<n;i++)
{for(j=i+1;j<n;j++)
{if(strcmp(s[j],s[i])<0)
{strcpy(temp,s[i]);
strcpy(s[i],s[j]);
strcpy(s[j],temp);}}
}
}

```

```

#include<stdio.h>
#include"stringA.c"
#include"stringB.c"
char s[30][30];
int n;
void search();
void main()
{int i;
flushall();
printf("\nEnter n value: ");
scanf("%d",&n);
getstring();
sort();
printf("\nSorted String array\n");
for(i=0;i<n;i++)
printf("%s\t",s[i]);
}

```

```

search();
getch();
}

void search()
{char stringToSearch[30];int i,counter=0;
printf("\nEnter the string to be searched: ");
gets(stringToSearch);
for(i=0;i<n;i++){
if(!strcmp(stringToSearch,s[i]))
counter++;}
if(counter==0)
printf("\nNO match found");
else
printf("\nTotal %d match(es) found.",counter);
}

```

105. 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 Equivalent : ");
while(mask > 0)
{ if((num & mask) == 0 )
printf("0");
else
printf("1");
mask = mask >> 1 ;
}
}

```

106. Find Largest 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); }

```

107.Matrix multiplication using dynamic memory allocation

```

#include <stdio.h>
#include<stdlib.h>
/* Main Function */
void main()
{ /* Declaring pointer fo 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");
for(j=0; j < col2; j++)
printf("%4d", ptr3[i][j]);}
return(0);
}

```

108.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);
}

```

109.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);
}

```

110.Addition without using +

```

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

```

111.Addition without using arithmetic operators

```

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

```

}