

"LEARN FROM THE MISTAKES OF OTHERS... YOU CAN'T LIVE LONG ENOUGH TO MAKE THEM ALL YOURSELVES!!"

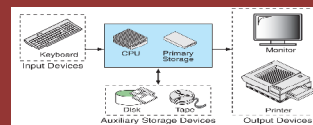
KNREDDY

PROGRAMMING IN C AND DATA STRUCTURES LAB PROGRAMS



చేయగలిగిన సత్యా ఉన్నా
చేయలేవేమోనన్న భయం
వెంటాడుతూ ఉంటుంది
భయం శక్తి వంతమైనదే
కాని నమ్మకం అంతకన్నా శక్తివంతమైనది
నమ్మకంతో ముందుకు వెళ్ళు
విజయం ఎప్పుడూ నీతోనే ఉంటుంది

K.NAGESWARA REDDY
ASSISTANT PROFESSOR



```
Preprocessor Directives

Global Declarations

int main (void)
{
    Local Declarations
    Statements
} // main

Other functions as required.
```

"Education is the Best Friend. An Educated Person is Respected Everywhere. Education beats the Beauty and the Youth."

(13A12102) PROGRAMMING IN C & DATA STRUCTURES LAB

LIST OF EXPERIMENTS/TASKS

1. Practice DOS and LINUX Commands necessary for design of C Programs.
2. Study of the Editors, Integrated development environments, and Compilers in chosen platform.
3. Write, Edit, Debug, Compile and Execute Sample C programs to understand the programming environment.
4. Practice programs: Finding the sum of three numbers, exchange of two numbers, maximum of two numbers, to read and print variable values of all data types of C language, to find the size of all data types, to understand the priority and associativity of operators using expressions, to use different library functions of C language.
5. Write a program to find the roots of a quadratic equation.
6. Write a program to compute the factorial of a given number.
7. Write a program to check whether the number is prime or not.
8. Write a program to find the series of prime numbers in the given range.
9. Write a program to generate Fibonacci numbers in the given range.
10. Write a program to find the maximum of a set of numbers.
11. Write a program to reverse the digits of a number.
12. Write a program to find the sum of the digits of a number.
13. Write a program to find the sum of positive and negative numbers in a given set of numbers.
14. Write a program to check for number palindrome.
15. Write a program to evaluate the sum of the following series up to 'n' terms
$$x=1+x+\frac{x^2}{2!}+\frac{x^3}{3!}+\frac{x^4}{4!}+\dots$$
16. Write a program to generate Pascal Triangle.
17. Write a program to read two matrices and print their sum and product in the matrix form.
18. Write a program to read matrix and perform the following operations.
 - i. Find the sum of Diagonal Elements of a matrix.
 - ii. Print Transpose of a matrix.
 - iii. Print sum of even and odd numbers in a given matrix.
19. Write a program to accept a line of characters and print the count of the number of Vowels, Consonants, blank spaces, digits and special characters.

20. Write a program to insert a substring in to a given string and delete few characters from the string. Don't use library functions related to strings.
21. Write a program to perform the operations addition, subtraction, multiplication of complex numbers.
22. Write a program to split a 'file' in to two files, say file1 and file2. Read lines into the 'file' from standard input. File1 should consist of odd numbered lines and file2 should consist of even numbered lines.
23. Write a program to merge two files.
24. Write a program to implement numerical methods Lagrange's interpolation, Trapezoidal rule.
25. Write a program to read a set of strings and sort them in alphabetical order.
26. Write a program to sort the elements of an array using sorting by exchange.
27. Write a program to sort the elements of an array using Selection Sort.
28. Write a program to perform Linear Search on the elements of a given array.
29. Write a program to perform Binary Search on the elements of a given array.
30. Write a program to find the number of occurrences of each number in a given array of numbers.
31. Write a program to read two strings and perform the following operations without using built-in string Library functions and by using your own implementations of functions.
 - i. String length determination
 - ii .Compare Two Strings
 - iii. Concatenate them, if they are not equal
 - iv. String reversing
32. Write programs using recursion for Factorial of a number, GCD, LCM, Towers of Hanoi.
33. Write a program to convert infix expression to postfix expression and evaluate postfix expression.
34. Write a program to exchange two numbers using pointers.
35. Write a program to implement stack, queue, circular queue using array and linked lists.
36. Write a program to perform the operations creation, insertion, deletion, and traversing a singly linked list
37. Write a program to read student records into a file. Record consists of rollno, name and marks of a student in six subjects and class. Class field is empty initially. Compute the class of a student. The calculation of the class is as per JNTUA rules. Write the first class, second class, third class and failed students lists separately to another file.
38. A file consists of information about employee salary with fields employeeid, name, Basic, HRA, DA, IT, other-deductions, Gross and Net salary. Initially only employeeid, name, and basic have valid values. HRA is taken as 10% of the basic, DA is taken as 80% of basic, IT is 20% of the

basic, other deductions is user specified. Compute the Gross and Net salary of the employee and update the file.

39. Write a program to perform Base (decimal, octal, hexadecimal, etc) conversion.
40. Write a program to find the square root of a number without using built-in library function.
41. Write a program to convert from string to number.
42. Write a program to generate pseudo random generator.
43. Write a program to remove duplicates from ordered and unordered arrays.
44. Write a program to sort numbers using insertion sort.
45. Write a program to implement quick sort using non-recursive and recursive approaches. Use randomized element as partitioning element.
46. Write a program to search a word in a given file and display all its positions.
47. Write a program to generate multiplication tables from 11 to 20.
48. Write a program to express a four digit number in words. For example 1546 should be written as one thousand five hundred and forty six.
49. Write a program to generate a telephone bill. The contents of it and the rate calculation etc should be as per BSNL rules. Student is expected to gather the required information through the BSNL website.
50. Write a program for tic-tac-toe game.
51. Write a program to find the execution time of a program.
52. Design a file format to store a person's name, address, and other information. Write a program to read this file and produce a set of mailing labels

References:

1. "Programming with C", Byron Gottfried, Third Edition, Schaum's Outlines, Mc Graw Hill.
2. "Fundamentals of Data Structures in C", Horowitz, Sahni, Anderson-freed, Second Edition, Universities Press.
3. "How to Solve it by Computer", R.G. Dromey, Pearson.
4. "The C Programming Language", Brian W. Kernighan, Dennis M. Ritchie, Pearson.
5. "Classic Data Structures", Samantha, PHI
6. "Let us C", Yeswant Kanetkar, BPB publications
7. "Pointers in C", Yeswant Kanetkar, BPB publications

INDEX

PROG NO	DATE	TITLE	PAGE NO
1		WRITE A C PROGRAM TO FIND SUM OF THREE NUMBERS	1
2		WRITE A C PROGRAM TO EXCHANG TWO NUMBERS	2
3		WRITE A C PROGRAM TO FIND MAXIMUM OF TWO NUMBERS	3
4.		WRITE A C PROGRAM TO FIND GIVEN NUMBER IS EVEN OR ODD	4
5		WRITE A C PROGRAM TO FIND ROOTS OF QUADRATIC EQUATION	5
6		WRITE A C PROGRAM TO COMPUTE FACTORIAL OF A GIVEN NUMBER	7
7		WRITE A C PROGRAM TO CHECK WHETHER THE NUMBER IS PRIME OR NOT	8
8		WRITE A C PROGRAM TO FIND THE SERIES OF PRIME NUMBERS IN THE GIVEN RANGE	9
9		WRITE A C PROGRAM TO GENERATE FIBONACCI NUMBERS IN THE GIVEN RANGE.	10
10		WRITE A C PROGRAM TO FIND THE MAXIMUM OF A SET OF NUMBERS	11
11		WRITE A C PROGRAM TO REVERSE THE DIGITS OF A NUMBER	12
12		WRITE A C PROGRAM TO FIND THE SUM OF THE DIGITS OF A NUMBER	13
13		WRITE A C PROGRAM TO FIND THE SUM OF POSITIVE AND NEGATIVE NUMBERS IN A GIVEN SET OF NUMBERS.	14
14		WRITE A C PROGRAM TO CHECK FOR NUMBER PALINDROME.	15
15		WRITE A C PROGRAM TO EVALUATE THE SUM OF THE FOLLOWING SERIES UP TO 'N' TERMS eX=1+X+X2/2!+X3/3!+X4/4!+-----	16
16.a		WRITE A C PROGRAM TO GENERATE PASCAL TRIANGLE USING BINOMIAL THEOREM	17
16.b		WRITE A C PROGRAM TO GENERATE PASCAL TRIANGLE USING ARRAYS.	19
17.a		WRITE A C PROGRAM TO READ TWO MATRICES AND PRINT THEIR SUM AND PRODUCT IN THE MATRIX FORM(WITHOUT USING FUNCTIONS)	20
17.b		WRITE A C PROGRAM TO READ TWO MATRICES AND PRINT THEIR SUM AND PRODUCT IN THE MATRIX FORM(USING FUNCTIONS)	23

PROG NO	DATE	TITLE	PAGE NO
18		WRITE A C PROGRAM TO READ MATRIX AND PERFORM THE FOLLOWING OPERATIONS. I. FIND THE SUM OF DIAGONAL ELEMENTS OF A MATRIX. II. PRINT TRANSPOSE OF A MATRIX. III. PRINT SUM OF EVEN AND ODD NUMBERS IN A GIVEN MATRIX	27
19		WRITE A C PROGRAM THAT USES FUNCTIONS TO PERFORM THE FOLLOWING OPERATIONS: I) READING A COMPLEX NUMBER II) WRITING A COMPLEX NUMBER III) ADDITION OF TWO COMPLEX NUMBERS IV) MULTIPLICATION OF TWO COMPLEX NUMBERS	30
20		WRITE A PROGRAM USING RECURSIVE FUNCTION FOR FINDING FACTORIAL OF A NUMBER	33
21		WRITE A PROGRAM USING RECURSIVE FUNCTION TOWERS OF HANOI	34
22		WRITE A PROGRAM FOR ARRAY ORDER REVERSAL	36
23		WRITE A PROGRAM TO REMOVE DUPLICATES FORM AN ARRAY	38
24		WRITE A PROGRAM TO FIND STRING LENGTH AND COPY STRINGS USING USERDEFINED FUNCTIONS	40
25		WRITE A PROGRAM TO COMPARE TWO STRINGS USING USER DEFINED PROGRAMS	42
26		WRITE A PROGRAM TO ACCEPT A LINE OF CHARACTERS AND PRINT THE COUNT OF THE NUMBER OF VOWELS, CONSONANTS, BLANK SPACES, DIGITS AND SPECIAL CHARACTERS.	44
27a		WRITE A C PROGRAM THAT USES FUNCTIONS TO PERFORM THE FOLLOWING OPERATIONS: TO INSERT A SUB-STRING IN TO GIVEN MAIN STRING FROM A GIVEN POSITION.	46
27b		TO DELETE N CHARACTERS FROM A GIVEN POSITION IN A GIVEN STRING.	48
28		WRITE A PROGRAM TO READ FIVE CITIES AND SORT THEM AND PRINT SORTED LIST OF CITIES IN ALPHABETICAL ORDER	50
29		WRITE A PROGRAM TO FIND WHETHER GIVEN STRING IS PALINDROME OR NOT	52
30		WRITE A C PROGRAM THAT DISPLAYS THE POSITION OR INDEX IN THE STRING S WHERE THE STRING T BEGINS, OR - 1 IF S DOESN'T CONTAIN T.	53
31		WRITE A C PROGRAM TO CONVERT A ROMAN NUMERAL TO ITS DECIMAL EQUIVALENT	54

PROG NO	DATE	TITLE	PAGE NO
32		WRITE A C PROGRAM TO FIND THE 2'S COMPLEMENT OF A BINARY NUMBER.	56
33		WRITE A C PROGRAM TO IMPLEMENT BUBBLE SORT (EXCHANGE SORT) METHODS TO SORT A GIVEN LIST OF INTEGERS IN ASCENDING ORDER	59
34		WRITE A C PROGRAM TO IMPLEMENT SELECTION SORT METHODS TO SORT A GIVEN LIST OF INTEGERS IN ASCENDING ORDER	61
35		WRITE A C PROGRAM TO IMPLEMENT INSERTION SORT METHODS TO SORT A GIVEN LIST OF INTEGERS IN ASCENDING ORDER	63
36		WRITE A C PROGRAM TO IMPLEMENT QUICK SORT METHODS TO SORT A GIVEN LIST OF INTEGERS IN ASCENDING ORDER	65
37		WRITE A C PROGRAM TO IMPLEMENT MERGE SORT METHODS TO SORT A GIVEN LIST OF INTEGERS IN ASCENDING ORDER	68
38a		WRITE A C PROGRAMS THAT USE NON RECURSIVE FUNCTIONS TO PERFORM LINEAR SEARCHING OPERATION FOR A KEY VALUE IN A GIVEN LIST OF INTEGERS	71
38b		WRITE A C PROGRAMS THAT USE RECURSIVE FUNCTIONS TO PERFORM LINEAR SEARCHING OPERATION FOR A KEY VALUE IN A GIVEN LIST OF INTEGERS	75
39a		WRITE A C PROGRAMS THAT USE NON RECURSIVE FUNCTIONS TO PERFORM BINARY SEARCHING OPERATION FOR A KEY VALUE IN A GIVEN LIST OF INTEGERS	78
39b		WRITE A C PROGRAMS THAT USE RECURSIVE FUNCTIONS TO PERFORM BINARY SEARCHING OPERATION FOR A KEY VALUE IN A GIVEN LIST OF INTEGERS	81
40		WRITE A PROGRAM TO TWO-WAY MERGE	84
41		WRITE A C PROGRAM TO EXCHANGE TWO NUMBERS USING POINTERS.	86
42		WRITE A C PROGRAM TO READ A LIST OF STUDENTS INFORMATION AND PRINT THEM IN ASCENDING ORDER OF THEIR AVERAGE MARKS USING ARRAY OF STRUCTURES	87
43		Write a C program to read student records into a file. Record consists of rollno, name and marks of a student in six subjects and class. Compute the class of a student. Write the first class, second class, third class and failed students lists separately to another file.	90

44		WRITE A C PROGRAM TO MERGE TWO FILES INTO A THIRD FILE	93
45		WRITE A C PROGRAMS THAT IMPLEMENT STACK (ITS OPERATIONS) USING ARRAYS	95
46		WRITE A C PROGRAMS THAT IMPLEMENT QUEUE (ITS OPERATIONS) USING ARRAYS	98
47		WRITE A C PROGRAM THAT USES STACK OPERATIONS TO PERFORM THE FOLLOWING: I) CONVERTING INFIX EXPRESSION INTO POSTFIX EXPRESSION II) EVALUATING THE POSTFIX EXPRESSION	101

1. WRITE A C PROGRAM TO FIND SUM OF THREE NUMBERS.

```
#include<stdio.h>
#include<conio.h>
void main( )
{
    int num1,num2,num3,sum;
    clrscr();
    printf("\nEnter any three numbers\n");
    scanf("%d%d%d",&num1,&num2,&num3);
    sum=num1+num2+num3;
    printf("\nSum is %d",sum);
    getch();
}
```

OUTPUT:

Enter any three numbers

2 3 4

Sum is 9

Enter any three numbers

50

40

80

Sum is 170

Enter any three numbers

11

22

33

Sum is 66

2. WRITE A C PROGRAM TO EXCHANG TWO NUMBERS

```
#include<stdio.h>
#include<conio.h>
void main( )
{
    int num1,num2,temp;
    clrscr( );
    printf("\nEnter any two numbers\n");
    scanf("%d%d",&num1,&num2);
    printf("\nNumbers before exchange are:");
    printf("\nnum1=%d\nnum2=%d",num1,num2);
    temp=num1;
    num1=num2;
    num2=temp;
    printf("\nNumbers after exchange are:");
    printf("\nnum1=%d\nnum2=%d",num1,num2);
    getch( );
}
```

OUTPUT:

```
Enter any two numbers
20
30
Numbers before exchange are:
num1=20
num2=30
Numbers after exchange are:
num1=30
num2=20
-----
Enter any two numbers
420
360
Numbers before exchange are:
num1=420
num2=360
Numbers after exchange are:
num1=360
num2=420
```

3. WRITE A C PROGRAM TO FIND MAXIMUM OF TWO NUMBERS

```
#include<stdio.h>
#include<conio.h>
void main( )
{
    int num1,num2;
    clrscr( );
    printf("Enter any two numbers\n");
    scanf("%d%d",&num1,&num2);
    if(num1>num2)
    {
        printf("%d is maximum of %d,%d",num1,num1,num2);
    }
    else
    {
        printf("%d is maximum of %d,%d",num2,num1,num2);
    }
    getch( );
}
```

OUTPUT:

Enter any two numbers

20

40

40 is maximum of 20, 40

Enter any two numbers

50

30

50 is maximum of 50, 30

4. WRITE A C PROGRAM TO FIND GIVEN NUMBER IS EVEN OR ODD

```
#include<stdio.h>
#include<conio.h>
void main( )
{
    int n;
    clrscr();
    printf("Enter the number : ");
    scanf("%d",&n);
    if (n%2==0)
    {
        printf("\nNumber is Even");
    }
    else
    {
        printf("\nNumber is Odd");
    }
    getch();
}
```

OUTPUT:

Enter the number : 120

Number is Even

Enter the number : 123

Number is Odd

5. WRITE A C PROGRAM TO FIND ROOTS OF QUADRATIC EQUATION

```
/* PROGRAM TO FIND ROOTS OF QUADRATIC EQUATION. */
```

```
/*Description: Nature of roots of quadratic equation ( $ax^2 + bx + c = 0$ ) can be known from the quadrant  $b^2 - 4ac$ 
```

$$\text{Roots} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

```
If  $b^2 - 4ac > 0$  then roots are real and unequal
```

```
If  $b^2 - 4ac = 0$  then roots are real and equal
```

```
If  $b^2 - 4ac < 0$  then roots are imaginary */
```

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

```
#include<conio.h>
```

```
#include<math.h>
```

```
void main( )
```

```
{
```

```
float a,b,c,root1,root2;
```

```
clrscr( );
```

```
printf("\n Enter values of a,b,c for finding roots of a quadratic eq:\n");
```

```
scanf("%f%f%f",&a,&b,&c);
```

```
if(b*b>4*a*c) /*checking condition*/
```

```
{
```

```
root1=(-b+sqrt(b*b-4*a*c))/(2*a);
```

```
root2=(-b-sqrt(b*b-4*a*c))/(2*a);
```

```
printf("\n*****ROOTS ARE REAL AND UNEQUAL*****\n");
```

```
printf("\n root1=%f\n root2=%f",root1,root2);
```

```
}
```

```
else if(b*b==4*a*c)
```

```
{
```

```
root1=-b/(2*a);
```

```
root2=-b/(2*a);
```

```
printf("\n*****ROOTS ARE REAL AND EQUAL*****\n");
```

```
printf("\n root1=%f\n root2=%f",root1,root2);
```

```
}
```

```
else
```

```
printf("\n Imaginary Roots.");
```

```
getch( );
```

```
}
```

OUTPUT:

Enter values of a,b,c for finding roots of a quadratic eq:

2 -5 2

*****ROOTS ARE REAL AND UNEQUAL*****

root1=2.000000

root2=0.500000

Enter values of a,b,c for finding roots of a quadratic eq:

2 3 4

Imaginary Roots.

Enter values of a,b,c for finding roots of a quadratic eq:

2 -4 2

*****ROOTS ARE REAL AND EQUAL*****

root1=1.000000

root2=1.000000

6. WRITE A C PROGRAM TO COMPUTE FACTORIAL OF A GIVEN NUMBER

```
#include<stdio.h>
#include<conio.h>
void main( )
{
    int n,i;
    long int fact=1;
    clrscr( );
    printf("Enter the number: ");
    scanf("%d",&n);
    if(n= =0)
        printf("Factorial of 0 is 1\n");
    else
    {
        for(i = 1; i <= n; i++)
        {
            fact=fact*i;
        }
        printf("Factorial of %d is %ld\n",n,fact);
    }
    getch( );
}
```

OUTPUT:

Enter the number: 5

Factorial of 5 is 120

Enter the number: 8

Factorial of 8 is 40320

Enter the number: 12

Factorial of 12 is 479001600

7. WRITE A C PROGRAM TO CHECK WHETHER THE NUMBER IS PRIME OR NOT

```
#include <stdio.h>
#include<conio.h>
void main( )
{
    int num,i,check=0;
    clrscr();
    printf("Enter a number ");
    scanf("%d",&num);
    for(i=2;i<num-1;i++)
    {
        if(num%i==0)
        {
            check++;    /*INCREMENT CHECK IF NO. IS NOT A PRIME NO.*/
            break;
        }
    }
    if(check == 0)
        printf("%d is a prime number ",num);
    else
        printf("%d is not a prime number ",num);
    getch( );
}
```

OUTPUT:

Enter a number 8

8 is not a prime number

Enter a number 89

89 is a prime number

8. WRITE A C PROGRAM TO FIND THE SERIES OF PRIME NUMBERS IN THE GIVEN RANGE

```
#include <stdio.h>
#include<conio.h>
main()
{
    int num,i,j,check;
    clrscr();
    printf("\nEnter a number up to which you have to find prime numbers : ");
    scanf("%d",&num);
    printf("\nTHE PRIME NUMBER SERIES B/W 1 TO %d :\n\n",num);
    for(i=1;i<=num;i++)
    {
        check = 0;
        //THIS LOOP WILL CHECK A NO TO BE PRIME NO. OR NOT.
        for(j=2;j<i-1 ;j++)
        {   if(i%j==0)
            {
                check++;    // INCREMENT CHECK IF NO. IS NOT A PRIME NO.
                break;
            }
        }
        if(check==0)
            printf("%d\t",i);
    }
    getch();
}
```

OUTPUT:

Enter a number up to which you have to find prime numbers : 100

THE PRIME NUMBER SERIES B/W 1 TO 100 :

```
1    2    3    5    7    11   13   17   19   23
29   31   37   41   43   47   53   59   61   67
71   73   79   83   89   97
```

9. WRITE A C PROGRAM TO GENERATE FIBONACCI NUMBERS IN THE GIVEN RANGE.

/* A Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. */

```
#include<stdio.h>
#include<conio.h>
void main( )
{
    int num1=0, num2=1,n,counter,fab;
    clrscr( );
    printf("\nENTER LENGTH OF SERIES (N) : ");
    scanf("%d",&n);
    printf("\nFIBONACCI SERIES:\n");
    if(n==0)
        printf("\nyou had entered number zero hence no series");
    else if(n==1)
        printf("%d",num1);
    else if(n==2)
        printf("%d%d",num1,num2);
    else
    {
        printf("\n%d\t%d",num1,num2);
        for(counter =3; counter<=n; counter++)
        {
            fab=num1+num2;
            printf("\t%d",fab);
            num1=num2;
            num2=fab;
        }
    }
    getch( );
}
```

OUTPUT:

ENTER LENGTH OF SERIES (N) : 10

FIBONACCI SERIES:

0 1 1 2 3 5 8 13 21 34

10. WRITE A C PROGRAM TO FIND THE MAX AND MIN OF A SET OF NUMBERS

```
#include<stdio.h>
#include<conio.h>
void main( )
{
    int a[20],i,n,min,max;
    clrscr( );
    printf("enter how many numbers in the set(<20):");
    scanf("%d",&n);
    printf("Enter the numbers in the set :\n");
    for(i=0;i<n;i++)          // read the elements of an array
        scanf("%d",&a[i]);
    min=a[0];
    max=a[0];
    for(i=0;i<n;i++)          // read the elements of an array
    {
        if(a[i]<min)           // check the condition for minimum value
            min=a[i];
        if(a[i]>max)//check the condition for maximum value
            max=a[i];
    }
    printf("maximum number in the set is:%d\n",max);
    printf("minimum number in the set is:%d\n",min);
    getch( );
}
```

OUTPUT:

enter how many numbers in the set(<20):5

Enter the numbers in the set :

4

5

1

2

8

maximum number in the set is:8

minimum number in the set is:1

11. WRITE A C PROGRAM TO REVERSE THE DIGITS OF A NUMBER.

```
/* PROGRAM TO REVERSE THE DIGITS OF A NUMBER */  
  
#include<stdio.h>  
  
#include<conio.h>  
  
void main( )  
{  
  
    int num,num1,i,rev=0;  
  
    clrscr();  
  
    printf("enter the number ");  
  
    scanf("%d",&num);  
  
    num1=num;  
  
    while(num!=0)  
    {  
  
        i=num%10;  
  
        rev=i+10*rev;  
  
        num=num/10;  
  
    }  
  
    printf("\nreverse of %d is: %d",num1,rev);  
  
    getch( );  
  
}
```

OUTPUT:

enter the number 1234

reverse of 1234 is: 4321

enter the number 3201

reverse of 3201 is: 1023

12. WRITE A C PROGRAM TO FIND THE SUM OF THE DIGITS OF A NUMBER.

```
#include<stdio.h>
#include<conio.h>
void main( )
{
    int num, i=1, sum=0,num1;
    clrscr( );
    printf("Enter the number whose digits are to be added: ");
    scanf("%d",&num);
    num1=num;
    if(num<0)
        printf("the number is not valid");
    else
    {
        while(num!=0)
        {
            i=num%10;
            sum=sum+i;
            num=num/10;
        }
        printf("Sum of the digits of %d is: %d",num1,sum);
    }
    getch( );
}
```

OUTPUT:

Enter the number whose digits are to be added: 123

Sum of the digits of 123 is: 6

Enter the number whose digits are to be added: 6305

Sum of the digits of 6305 is: 14

Enter the number whose digits are to be added: 9996

Sum of the digits of 9996 is: 33

13. WRITE A C PROGRAM TO FIND THE SUM OF POSITIVE AND NEGATIVE NUMBERS IN A GIVEN SET OF NUMBERS.

```
#include<stdio.h>
#include<conio.h>
void main( )
{
    int a[20],i,n,sum_pos=0,sum_neg=0;
    clrscr();
    printf("enter how many numbers in the set(<20):");
    scanf("%d",&n);
    printf("Enter the numbers(both positive and negative) :\n");
    for(i=0;i<n;i++)    // read the elements of an array
        scanf("%d",&a[i]);
    for(i=0;i<n;i++)    // read the elements of an array
    {
        if(a[i]<0)    // check the condition for negative numbers
            sum_neg=sum_neg+a[i];
        else    // positive numbers
            sum_pos=sum_pos+a[i];
    }
    printf("sum of positive numbers in the set is: %d\n",sum_pos);
    printf("sum of negaive numbers in the set is: %d\n",sum_neg);
    getch( );
}
```

OUTPUT:

```
enter how many numbers in the set(<20):10
Enter the numbers(both positive and negative ):
2    -3    4    -4    5    -1    7    -8    9    8
sum of positive numbers in the set is: 35
sum of negaive numbers in the set is: -16
```

14. WRITE A C PROGRAM TO CHECK FOR NUMBER PALINDROME.

```
#include<stdio.h>
#include<conio.h>
void main( )
{
    int num,num1,i,rev=0;
    clrscr();
    printf("enter the number ");
    scanf("%d",&num);
    num1=num;
    while(num!=0)
    {
        i=num%10;
        rev=i+10*rev;
        num=num/10;
    }
    if(num1==rev)
        printf("\nGiven number %d is palindrome",num1);
    else
        printf("\nGiven number %d is not palindrome",num1);
    getch( );
}
```

OUTPUT:

enter the number 6305

Given number 6305 is not palindrome

enter the number 12321

Given number 12321 is palindrome

15. WRITE A C PROGRAM TO EVALUATE THE SUM OF THE FOLLOWING SERIES**UP TO 'N' TERMS**

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots$$

```

#include <stdio.h>
#include <conio.h>
#include <math.h>
void main( )
{
    int f,n;
    float ex=0,x,power,fact;
    clrscr( );
    printf("\n\nEQUATION SERIES : 1+x+X^2/2!+X^3/3!+X^4/4! -----");
    printf("\n\nENTER VALUE OF X : ");
    scanf("%f",&x);
    printf("\n\nENTER VALUE OF n : ");
    scanf("%d",&n);
    for(power=0; power<=n; power++)
    {
        fact=1;
        //CALC FACTORIAL OF POWER VALUE
        for(f=1;f<=power;f++)
            fact = fact*f;
        //EQ. FOR SUM SERIES
        ex=ex+(pow(x,power)/fact);
    }
    printf("\n\nResult: e^x= %f",ex);
    getch( );
}

```

OUTPUT:

EQUATION SERIES : 1+x+X^2/2!+X^3/3!+X^4/4! -----

ENTER VALUE OF X : 2

ENTER VALUE OF n : 3

Result: e^x= 6.333333

16A. WRITE A C PROGRAM TO GENERATE PASCAL TRIANGLE USING BINOMIAL THEOREM.

```
#include<stdio.h>
#include<conio.h>
void main( )
{
    int bin,p,q,r,x;
    clrscr( );
    bin=1;
    q=0;
    printf("How many 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\n");
        ++q;
    }
    getch( );
}
```

ROLL NO:

Page 18

OUTPUT:

How many rows you want to input:5

Pascal's Triangle:

```
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
```

16B. WRITE A C PROGRAM TO GENERATE PASCAL TRIANGLE USING ARRAYS.

```
#include<stdio.h>
#include<conio.h>
void main( )
{
    int i,j,position=30,line,a[20][20];
    clrscr( );
    printf("Enter number of lines to print pascal triangle\n");
    scanf("%d",&line);
    for(i=0;i<line;i++)
    {
        gotoxy(position,i+2);
        for(j=0;j<=i;j++)
        {
            if(j==0||i==j)
                a[i][j]=1;
            else
                a[i][j]=a[i-1][j-1]+a[i-1][j];
            printf("%6d",a[i][j]);
        }
        position=position-3;
        printf("\n\n");
    }
    getch( );
}
```

OUTPUT:

Enter number of lines to print pascal triangle

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

17A. WRITE A C PROGRAM TO READ TWO MATRICES AND PRINT THEIR SUM AND PRODUCT IN THE MATRIX FORM.(WITH OUT USING FUNCTIONS)

```
#include<stdio.h>
#include<conio.h>
void main( )
{
    int i,j,m,n,p,q,k,a[10][10],b[10][10],c[10][10];
    clrsrc( );
    printf("Input rows and columns of A matrix:");
    scanf("%d%d",&m,&n);
    printf("Input rows and columns of B matrix:");
    scanf("%d%d",&p,&q);
    printf("Enter elements of matrix A:\n");
    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
            scanf("%d",&a[i][j]);
    printf("Enter elements of matrix B:\n");
    for(i=0;i<p;i++)
        for(j=0;j<q;j++)
            scanf("%d",&b[i][j]);
    if(m==p&&n==q)
    {
        printf("\nMATRIX ADDITION IS POSSIBLE");
        printf("\nMATRIX ADDITION :\n");
        for(i=0;i<p;i++)
        {
            for(j=0;j<q;j++)
                c[i][j]=a[i][j]+b[i][j];
        }
        for(i=0;i<m;i++)
        {
            for(j=0;j<n;j++)
                printf("%5d",c[i][j]);
            printf("\n");
        }
    }
}
```

```
else
    printf("\nMATRIX ADDITION IS NOT POSSIBLE");
if(n==p)
{
    printf("\nMATRICES CAN BE MULTIPLIED");
    printf("\nRESULTANT MATRIX is %d*%d",m,q);
    printf("\nMATRIX MULTIPLICATION\n");
    for(i=0;i<m;++i)
        for(j=0;j<q;++j)
            {
                c[i][j]=0;
                for(k=0;k<n;++k)
                    c[i][j]=c[i][j]+a[i][k]*b[k][j];
            }
    for(i=0;i<m;i++)
        {
            for(j=0;j<q;j++)
                printf("%5d",c[i][j]);
            printf("\n");
        }
    }
else
    {
        printf("\nMATRIX MULTIPLICATION IS NOT POSSIBLE.");
    }
getch( );
}
```

OUTPUT:

Input rows and columns of A matrix:2 2

Input rows and columns of B matrix:2 2

Enter elements of matrix A:

1 1 1 1

Enter elements of matrix B:

2 2 2 2

MATRIX ADDITION IS POSSIBLE

MATRIX ADDITION :

3 3

3 3

MATRICES CAN BE MULTIPLIED

RESULTANT MATRIX is 2*2

MATRIX MULTIPLICATION

4 4

4 4

Input rows and columns of A matrix:2 3

Input rows and columns of B matrix:3 1

Enter elements of matrix A:

1 2 3 4 5 6

Enter elements of matrix B:

1 2 3

MATRIX ADDITION IS NOT POSSIBLE

MATRICES CAN BE MULTIPLIED

RESULTANT MATRIX is 2*1

MATRIX MULTIPLICATION

14

32

17B. WRITE A C PROGRAM TO READ TWO MATRICES AND PRINT THEIR SUM AND PRODUCT IN THE MATRIX FORM.(USING FUNCTIONS)

```
#include<stdio.h>
#include<conio.h>
int read_matrix(int a[10][10],int m,int n);
int write_matrix(int a[10][10],int m,int n);
void main( )
{
    int ch,i,j,m,n,p,q,k,r1,c1,a[10][10],b[10][10],c[10][10];
    char yn;
    clrscr();
    printf("*****");
    printf("\n\t\tMENU");
    printf("\n*****");
    printf("\n[1]ADDITION OF TWO MATRICES");
    printf("\n[2]MULTIPLICATION OF TWO MATRICES");
    printf("\n[0]EXIT");
    printf("\n*****");
    while(1)
    {
        printf("\n\tEnter your choice:\n");
        scanf("%d",&ch);
        if(ch<=2&&ch>0)
        {
            printf("Valid Choice\n");
        }
        switch(ch)
        {
            case 1: printf("Input rows and columns of A & B Matrix:");
                    scanf("%d%d",&r1,&c1);
                    printf("Enter elements of matrix A:\n");
                    read_matrix(a,r1,c1);
                    printf("Enter elements of matrix B:\n");
                    /*Function call to read the matrix*/
                    read_matrix(b,r1,c1);
                    printf("\n =====Matrix Addition===== \n");
                    PROGRAMMING IN C AND DATA STRUCTURES
                    KNREDDY
```

```

        for(i=0;i<r1;i++)
        {
            for(j=0;j<c1;j++)
                c[i][j]=a[i][j]+b[i][j];
        }
        /*Function call to write the matrix*/
        write_matrix(c,r1,c1);
        break;
case 2: printf("Input rows and columns of A matrix:");
        scanf("%d%d",&m,&n);
        printf("Input rows and columns of B matrix:");
        scanf("%d%d",&p,&q);
        if(n==p)
        {
            printf("matrices can be multiplied\n");
            printf("resultant matrix is %d*%d\n",m,q);
            printf("Input A matrix\n");
            read_matrix(a,m,n);
            printf("Input B matrix\n");
            /*Function call to read the matrix*/
            read_matrix(b,p,q);
            printf("\n =====Matrix Multiplication===== \n");
            for(i=0;i<m;++i)
                for(j=0;j<q;++j)
                {
                    c[i][j]=0;
                    for(k=0;k<n;++k)
                        c[i][j]=c[i][j]+a[i][k]*b[k][j];
                }
            printf("Result of product two matrices:\n");
            /*Function call to write the matrix*/
            write_matrix(c,m,q);
        } /*end if*/
    else
    {
        printf("Matrices cannot be multiplied.");
    } /*end else*/

```



```
        break;
    case 0:    printf("\n Choice Terminated");
              break;
    default:  printf("\n Invalid Choice");
              } /* end of switch*/
    printf("DO YOU WANT TO DO ANOTHER OPERATION(Y/N):");
    scanf(" %c",&yn);
    if(yn=='n'||yn=='N')
        break;
} /* end of while */
getch( );
}
/*Function read matrix*/
int read_matrix(int a[10][10],int m,int n)
{
    int i,j;
    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
            scanf("%d",&a[i][j]);
    return 0;
}
/*Function to write the matrix*/
int write_matrix(int a[10][10],int m,int n)
{
    int i,j;
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
            printf("%5d",a[i][j]);
        printf("\n");
    }
    return 0;
}
```

OUTPUT:

MENU

[1]ADDITION OF TWO MATRICES

[2]MULTIPLICATION OF TWO MATRICES

[0]EXIT

Enter your choice:

1

Valid Choice

Input rows and columns of A & B Matrix:2 2

Enter elements of matrix A:

1 1 1 1

Enter elements of matrix B:

2 2 2 2

=====Matrix Addition=====

3 3

3 3

DO YOU WANT TO DO ANOTHER OPERATION(Y/N):y

Enter your choice:

2

Valid Choice

Input rows and columns of A matrix:2 2

Input rows and columns of B matrix:2 2

matrices can be multiplied

resultant matrix is 2*2

Input A matrix

1 1 1 1

Input B matrix

2 2 2 2

=====Matrix Multiplication=====

Result of product two matrices:

4 4

4 4

DO YOU WANT TO DO ANOTHER OPERATION(Y/N):n

18. WRITE A C PROGRAM TO READ MATRIX AND PERFORM THE FOLLOWING OPERATIONS.

I. FIND THE SUM OF DIAGONAL ELEMENTS OF A MATRIX.

II. PRINT TRANSPOSE OF A MATRIX.

III. PRINT SUM OF EVEN AND ODD NUMBERS IN A GIVEN MATRIX.

```
#include<stdio.h>

#include<conio.h>

void main( )

{

    int i,j,row,col,diagsum=0,evensum=0,oddsun=0,a[10][10],t[10][10];

    clrscr( );

    printf("Input rows and columns of matrix:");

    scanf("%d%d",&row,&col);

    printf("Enter elements of matrix :\n");

    for(i=0;i<row;i++)

    {

        for(j=0;j<col;j++)

            scanf("%d",&a[i][j]);

    }

    printf("GIVEN MATRIX IS:\n");

    for(i=0;i<row;i++)

    {

        for(j=0;j<col;j++)

            printf("%5d",a[i][j]);

        printf("\n");

    }

    printf("\n[1].SUM OF DIAGONAL ELEMENTS OF GIVEN MATRIX : ");

    for(i=0;i<row;i++)

        for(j=0;j<col;j++)

            if(i==j)
```

```
        diagsum=diagsum+a[i][j];

printf("%d",diagsum);

printf("\n[2].TRANPOSE OF GIVEN MATRIX : \n");

for(i=0;i<col;i++)

{

    for(j=0;j<row;j++)

        printf("%5d",a[j][i]);

    printf("\n");

}

printf("\n[3].SUM OF EVEN AND ODD NUMBERS IN A GIVEN MATRIX: ");

for(i=0;i<row;i++)

{

    for(j=0;j<col;j++)

    {

        if(a[i][j]%2==0)

            evensum=evensum+a[i][j];

        else

            oddsum=oddsum+a[i][j];

    }

}

printf("\nSUM OF EVEN NUMBERS IN GIVEN MATRIX: %d",evensum);

printf("\nSUM OF ODD NUMBERS IN GIVEN MATRIX: %d",oddsum);

getch( );

}
```

ROLL NO:

Page 29

OUTPUT:

Input rows and columns of matrix:2 3

Enter elements of matrix :

1 2 3 4 5 6

GIVEN MATRIX IS:

1 2 3

4 5 6

[1].SUM OF DIAGONAL ELEMENTS OF GIVEN MATRIX : 6

[2].TRANSPOSE OF GIVEN MATRIX :

1 4

2 5

3 6

[3].SUM OF EVEN AND ODD NUMBERS IN A GIVEN MATRIX:

SUM OF EVEN NUMBERS IN GIVEN MATRIX: 12

SUM OF ODD NUMBERS IN GIVEN MATRIX: 9

19. Write a C program that uses functions to perform the following operations:

- i) Reading a complex number**
- ii) Writing a complex number**
- iii) Addition of two complex numbers**
- iv) Multiplication of two complex numbers**

(Note: represent complex number using a structure.) */

```
#include<stdio.h>

#include<math.h>

void arithmetic(int opern);

struct comp
{
    double realpart;
    double imgpart;
};

void main( )
{
    int opern;

    clrscr ( );

    printf("\n\n \t\t\t***** MAIN MENU *****");

    printf("\n\n Select your option: \n 1 : ADD\n 2 : MULTIPLY\n 0 : EXIT \n\n\t\t");

    printf("\n Enter your Option [ 0-2 ]\n");

    scanf("%d",&opern);

    switch(opern)
    {
        case 0:
            exit(0);

        case 1:

        case 2:

            arithmetic(opern);
```

```
        default:

            main( );

        }

    }

void arithmetic(int opern)

{

    struct comp w1, w2, w;

    printf("\n Enter two Complex Numbers (x+iy):\n Real Part of First Number:");

    scanf("%lf",&w1.realpart);

    printf("\n Imaginary Part of First Number:");

    scanf("%lf",&w1.imgpart);

    printf("\n Real Part of Second Number:");

    scanf("%lf",&w2.realpart);

    printf("\n Imaginary Part of Second Number:");

    scanf("%lf",&w2.imgpart);

    switch(opern)

    {

        /*addition of complex number*/

        case 1:

            w.realpart = w1.realpart+w2.realpart;

            w.imgpart = w1.imgpart+w2.imgpart;

            break;

        /*multiplication of complex number*/

        case 2:

            w.realpart=(w1.realpart*w2.realpart)-(w1.imgpart*w2.imgpart);

            w.imgpart=(w1.realpart*w2.imgpart)+(w1.imgpart*w2.realpart);

            break;

    }

}
```

```
if (w.imgpart>0)
    printf("\n Answer = %lf+%lfi",w.realpart,w.imgpart);
else
    printf("\n Answer = %lf%lfi",w.realpart,w.imgpart);
getch();
main( );
}
```

OUTPUT:

***** MAIN MENU *****

Select your option:

1 : ADD

2 : MULTIPLY

0 : EXIT

Enter your Option [1]

Enter two Complex Numbers (x+iy):

Real Part of First Number:2

Imaginary Part of First Number:2

Real Part of Second Number:2

Imaginary Part of Second Number:2

Answer = 4.000000+4.000000i

20. WRITE A PROGRAM USING RECURSIVE FUNCTION FOR FINDING FACTORIAL OF A NUMBER

```
#include<stdio.h>

#include<conio.h>

unsigned int factorial(int n);

void main( )

{

    int n;

    printf("Enter the number: ");

    scanf("%d",&n);

    printf("Factorial of %d Using Recursive Function is %d\n",n,factorial(n));

    getch( );

}

/* Recursive Function*/

unsigned int factorial(int n)

{

    return(n>=1 ? n*factorial(n-1): 1);

}
```

OUTPUT:

Enter the number: 5

Factorial of 5 Using Recursive Function is 120

21. WRITE A PROGRAM USING RECURSIVE FUNCTION TOWERS OF HANOI

```
#include<conio.h>

#include<stdio.h>

#include<math.h>

void hanoiRecursion( int num,char st, char dt, char it);

void main( )

{

    int no,unsigned long int n;

    clrscr( );

    printf("Enter the no. of disks to be transferred: ");

    scanf("%d",&no);

    if(no<1)

        printf("\nThere's nothing to move.");

    else

    {

        n=pow(2,no)-1;

        printf("\nNumber of steps required for moving %d disks are: %lu",no,n);

        printf("\nEnter any key to display moves");

        getch();

        printf("\nRecursive Function");

        printf("\n-----");

        hanoiRecursion(no,'A','C','B');

    }

    getch( );

}

/* Recursive Function*/

void hanoiRecursion( int num,char st, char dt, char it)

{

    if ( num == 1 )
```

```
{  
    printf( "\nMove top disk from needle %c to needle %c.", st, dt );  
    return;  
}  
hanoiRecursion( num - 1,st, it, dt );  
printf( "\nMove top disk from needle %c to needle %c.", st, dt );  
hanoiRecursion( num - 1,it, dt, st );  
}
```

OUTPUT:

Enter the no. of disks to be transferred: 3

Number of steps required for moving 3 disks are: 7

Enter any key to display moves

Recursive Function

Move top disk from needle A to needle C.

Move top disk from needle A to needle B.

Move top disk from needle C to needle B.

Move top disk from needle A to needle C.

Move top disk from needle B to needle A.

Move top disk from needle B to needle C.

Move top disk from needle A to needle C.

22. WRITE A PROGRAM FOR ARRAY ORDER REVERSAL

```
#include<stdio.h>

#include<conio.h>

void main( )
{
    int a[100],i,r,t,n;

    clrscr( );

    printf("\nEnter number of elements to enter into array");

    scanf("%d",&n);

    printf("\nEnter any %d numbers",n);

    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }

    printf("\nElements in array before reversal\n");

    for(i=0;i<n;i++)
        printf("a[%d]=%d\t",i,a[i]);

    r=n/2;

    for(i=0;i<r;i++)
    {
        t=a[i];
        a[i]=a[n-1-i];
        a[n-1-i]=t;
    }

    printf("\nElements in array after reversal\n");

    for(i=0;i<n;i++)
        printf("a[%d]=%d\t",i,a[i]);

    getch( );
}
```

ROLL NO:

OUTPUT:

Enter number of elements to enter into array5

Enter any 5 numbers 1 2 3 4 5

Elements in array before reversal

a[0]=1 a[1]=2 a[2]=3 a[3]=4 a[4]=5

Elements in array after reversal

a[0]=5 a[1]=4 a[2]=3 a[3]=2 a[4]=1

23. WRITE A PROGRAM TO REMOVE DUPLICATES FORM AN ARRAY

```
#include<stdio.h>

#include<conio.h>

void main( )

{

    int a[20],i,j,k,n;

    clrscr( );

    printf("\nEnter array size : ");

    scanf("%d",&n);

    printf("\nAccept Numbers :\n");

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

        scanf("%d",&a[i]);

    printf("\nOriginal array is:\n");

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

        printf("%d ",a[i]);

    printf("\nUpdated array is:\n");

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

    {

        for(j=i+1;j<n;)

        {

            if(a[j]==a[i])

            {

                for(k=j;k<n;k++)

                    a[k]=a[k+1];

                n--;

            }

            else

                j++;

        }

    }
```

```
}  
  
for(i=0;i<n;i++)  
    printf("%d ",a[i]);  
  
getch( );  
  
}
```

OUTPUT:

Enter array size : 5

Accept Numbers :

1 2 2 3 4

Original array is:

1 2 2 3 4

Updated array is:

1 2 3 4

24. WRITE A PROGRAM TO FIND STRING LENGTH AND COPY STRINGS USING USERDEFINED FUNCTIONS

```
#include<stdio.h>

#include<conio.h>

int strlenh(char str[])

{

    int count=0;

    while(str[count]!='\0')

        count++;

    return(count);

}

void strcpy(char dest[],char src[])

{

    int i,length;

    length=strlength(src);

    for(i=0;i<length;i++)

        dest[i]=src[i];

    dest[length]='\0';

}

void main( )

{

    int len;

    char str1[100],str2[100];

    clrscr( );

    printf("\nEnter a string :");

    gets(str1);

    len=strlength(str1);

    printf("\nThe length of string is %d",len);

    strcpy(str2,str1);
```



```
printf("\nSource string is %s\nCopied string is %s",str1,str2);  
  
getch();  
  
}
```

OUTPUT:

Enter a string :c program

The length of string is 9

Source string is c program

Copied string is c program

25. WRITE A PROGRAM TO COMPARE TWO STRINGS USING USER DEFINED PROGRAMS

```
#include<stdio.h>

#include<conio.h>

int strlenh(char str[])
{
    int count=0;

    while(str[count]!='\0')
        count++;

    return(count);
}

int strcmpare(char str1[],char str2[])
{
    int i,length,length1,length2;

    length1=strlength(str1);
    length2=strlength(str2);
    length=(length1<length2)?length1:length2;

    for(i=0;i<=length;i++)
        if(str1[i]<str2[i])
            return -1;
        else if(str1[i]>str2[i])
            return 1;

    return 0;
}

void main( )
{
    int status;

    char str1[100],str2[100];

    clrscr( );
```

```
printf("\nEnter first string");  
  
gets(str1);  
  
printf("\nEnter second string");  
  
gets(str2);  
  
status=strcompare(str1,str2);  
  
if(status==-1)  
  
    printf("\nFirst string is less than second string");  
  
else if(status==1)  
  
    printf("\nFirst string is greater than second string");  
  
else  
  
    printf("\nBoth strings are equal");  
  
getch( );  
  
}
```

26. WRITE A PROGRAM TO ACCEPT A LINE OF CHARACTERS AND PRINT THE COUNT OF THE NUMBER OF VOWELS, CONSONANTS, BLANK SPACES, DIGITS AND SPECIAL CHARACTERS.

```
/* program to count number of vowels , number of special characters and number of consonants in a given string*/
```

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

```
#include<conio.h>
```

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

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

```
void main( )
```

```
{
```

```
    char string[100],ch;
```

```
    int nchr=0,nv=0,nc=0,nsch=0,i=0;
```

```
    clrscr( );
```

```
    printf("Enter any string:");
```

```
    gets(string);
```

```
    while(string[i]!='\0')
```

```
    {
```

```
        ch=string[i];
```

```
        if(isalpha(ch))
```

```
        {
```

```
            nchr++;
```

```
            if(tolower(ch)=='a' || tolower(ch)=='e' || tolower(ch)=='i' || tolower(ch)=='o' || tolower(ch)=='u')
```

```
                nv++;
```

```
            else
```

```
                nc++;
```

```
        }
```

```
    else
```

```
    {
```

```
        nsch++;
```

```
        nchr++;  
    }  
    i++;  
}  
  
printf("\nNumber of special characters(including space) in the given string are: %d",nsch);  
printf("\nNumber of vowels in the given string are: %d",nv);  
printf("\nNumber of consonants in the given string are: %d",nc);  
printf("\nTotal Number of characters in the given string are: %d",nchr);  
getch( );  
}
```

OUTPUT:

Enter any string:this program counts vowels consoants and special characters

Number of special characters(including space) in the given string are: 7

Number of vowels in the given string are: 12

Number of consonants in the given string are: 40

Total Number of characters in the given string are: 59

27a. WRITE A C PROGRAM THAT USES FUNCTIONS TO PERFORM THE FOLLOWING OPERATIONS:

TO INSERT A SUB-STRING IN TO GIVEN MAIN STRING FROM A GIVEN POSITION.

```
#include <stdio.h>

#include <conio.h>

#include <string.h>

void main( )
{
    char a[10],b[10],c[10],x;

    int p=0,lena,lenb,i,t=0,lenab,k;

    clrscr( );

    puts("Enter First String:");

    gets(a);

    puts("Enter Second String:");

    gets(b);

    printf("Enter the position where the item has to be inserted: ");

    scanf("%d",&p);

    lena = strlen(a);

    lenb = strlen(b);

    /*Copying the input string into another array */

    for(i=0;i<=lena;i++)

    {

        c[i]=a[i];

    }

    lenab= lena+lenb;

    k = p-1+lenb;

    /*Adding the sub-string*/

    for(i=p-1;i<lenab;i++)

    {
```

```
x = c[i];  
  
if(t<lenb)  
{  
    a[i] = b[t];  
    t=t+1;  
}  
a[k]=x;  
k=k+1;  
}  
puts("Resultant string is:");  
puts(a);  
getch( );  
}
```

OUTPUT:

Enter First String:

c class

Enter Second String:

programming

Enter the position where the item has to be inserted: 2

Resultant string is:

cprogramming class

27b. WRITE A C PROGRAM THAT USES FUNCTIONS TO PERFORM THE FOLLOWING OPERATIONS:

TO DELETE N CHARACTERS FROM A GIVEN POSITION IN A GIVEN STRING.

```
#include <stdio.h>

#include <conio.h>

#include <string.h>

void delchar(char *x,int a, int b);

void main( )

{

    char string[10];

    int n,pos,p;

    clrscr( );

    puts("Enter the string");

    gets(string);

    printf("Enter the position from where to delete: ");

    scanf("%d",&pos);

    printf("Enter the number of characters to be deleted: ");

    scanf("%d",&n);

    delchar(string, n,pos);

    getch( );

}

void delchar(char *x,int a, int b)

{

    if ((a+b-1) <= strlen(x))

    {

        strcpy(&x[b-1],&x[a+b-1]);

        puts(x);

    }

}
```


ROLL NO:

Page 49

OUTPUT:

Enter the string

programming

Enter the position from where to delete: 5

Enter the number of characters to be deleted: 7

prog

28. WRITE A PROGRAM TO READ FIVE CITIES AND SORT THEM AND PRINT SORTED LIST OF CITIES IN ALPHABETICAL ORDER

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    char city[5][20],temp[20];
    int i,n=0;
    clrscr();
    printf("enter the names of five cities...\n\n");
    for(i=0;i<5;i++)
        scanf("%s",&city[i]);
    printf("sorted list of cities...\n\n");
    while(!n)
    {
        n=1;
        for(i=0;i<4;i++)
        {
            if(strcmp(city[i],city[i+1])>0)
            {
                n=0;
                strcpy(temp,city[i]);
                strcpy(city[i],city[i+1]);
                strcpy(city[i+1],temp);
            }
        }
    }
    for(i=0;i<5;i++)
        printf("\n%s",city[i]);
    getch();
}
```

ROLL NO:

Page 51

OUTPUT:

enter the names of five cities...

nandyal

hyderabad

delhi

guntur

bombay

sorted list of cities...

bombay

delhi

guntur

hyderabad

nandyal

29. WRITE A PROGRAM TO FIND WHETHER GIVEN STRING IS PALINDROME OR NOT

```
#include<stdio.h>

#include<conio.h>

#include<string.h>

void main( )

{

    char str[20];

    int len,flag=1,i,length;

    clrscr( );

    printf("Enter a string : ");

    scanf("%s",str);

    len = strlen(str);

    length=len-1;

    for(i=0;i<len/2;i++)

    {

        if(str[i]!=str[length-i])

            flag = 0;

    }

    if(flag==1)

        printf("\n Given string %s is palindrome",str);

    else

        printf("\n Given string %s is not a palindrome",str);

    getch( );

}
```

OUTPUT:

Enter a string : knreddy

Given string knreddy is not a palindrome

Enter a string : abcba

Given string abcba is palindrome

30. WRITE A C PROGRAM THAT DISPLAYS THE POSITION OR INDEX IN THE STRING S WHERE THE STRING T BEGINS, OR - 1 IF S DOESN'T CONTAIN T.

```
#include<stdio.h>

#include<string.h>

#include<conio.h>

void main( )

{

    char s[30], t[20];

    char *found;

    clrscr( );

    /* Entering the main string */

    puts("Enter the first string: ");

    gets(s);

    /* Entering the string whose position or index to be displayed */

    puts("Enter the string to be searched: ");

    gets(t);

    /*Searching string t in string s */

    found=strstr(s,t);

    if(found)

        printf("Second String \"%s\" is found in the First String \"%s\" and begins at %d
position.\n",t,s,found-s+1);

    else

        printf("-1");

    getch( );

}
```

OUTPUT:

Enter the first string: knreddy

Enter the string to be searched: red

Second String "red" is found in the First String "knreddy" and begins at 3 position.

31. WRITE A C PROGRAM TO CONVERT A ROMAN NUMERAL TO ITS DECIMAL EQUIVALENT.

```
#include<stdio.h>

#include<conio.h>

#include<string.h>

#include<stdlib.h>

void main()

{

    int a[10],len,i,j,k;

    char rom[10];

    clrscr();

    printf("Enter the Roman Numeral:");

    gets(rom);

    len=strlen(rom);

    for(i=0;i<len;i++)

    {

        if(rom[i]=='I'||rom[i]=='i')

            a[i]=1;

        else if(rom[i]=='V'||rom[i]=='v')

            a[i]=5;

        else if(rom[i]=='X'||rom[i]=='x')

            a[i]=10;

        else if(rom[i]=='L'||rom[i]=='l')

            a[i]=50;

        else if(rom[i]=='C'||rom[i]=='c')

            a[i]=100;

        else if(rom[i]=='D'||rom[i]=='d')

            a[i]=500;
```

```
        else if(rom[i]=='M'||rom[i]=='m')
            a[i]=1000;
        else
        {
            printf("\nInvalid Value");
            getch();
            exit(0);
        }
    }
    k=a[len-1];
    for(i=len-1;i>0;i--)
    {
        if(a[i]>a[i-1])
            k=k-a[i-1];
        else if(a[i]==a[i-1] || a[i]<a[i-1])
            k=k+a[i-1];
    }
    printf("\nIts Decimal Equivalent is:");
    printf("%d",k);
    getch();
}
```

OUTPUT:

Enter the Roman Numeral: xxx

Its Decimal Equivalent is: 30

32. 2's complement of a number is obtained by scanning it from right to left and complementing all the bits after the first appearance of a 1. Thus 2's complement of 11100 is 00100.

Write a C program to find the 2's complement of a binary number.

```
#include <stdio.h>

#include<conio.h>

void complement (char *a);

void main( )
{
    char a[16];
    int i;
    clrscr( );
    printf("Enter the binary number");
    gets(a);
    for(i=0;a[i]!='\0'; i++)
    {
        if (a[i]!='0' && a[i]!='1')
        {
            printf("The number entered is not a binary number. Enter the correct number");
            exit(0);
        }
    }
    complement(a);
    getch( );
}

void complement (char *a)
{
    int len, i, c=0;
    char b[16];
    len=strlen(a);
```



```
for (i=len-1; i>=0; i--)
{
    if (a[i]=='0')
        b[i]='1';
    else
        b[i]='0';
}
for(i=len-1; i>=0; i--)
{
    if(i==len-1)
    {
        if (b[i]=='0')
            b[i]='1';
        else
        {
            b[i]='0';
            c=1;
        }
    }
    else
    {
        if(c==1 && b[i]=='0')
        {
            b[i]='1';
            c=0;
        }
        else if (c==1 && b[i]=='1')
        {
            b[i]='0';
```

```
        c=1;
    }
}
}
b[len]='\0';
printf("The 2's complement is %s", b);
}
```

OUPUT:

Enter the binary number1010

The 2's complement is 0110

33. WRITE A C PROGRAM TO IMPLEMENT BUBBLE SORT (EXCHANGE SORT) METHODS TO SORT A GIVEN LIST OF INTEGERS IN ASCENDING ORDER

```
#include<stdio.h>

#include<conio.h>

#define MAX 20

void readlist(int list[ ],int n)

{

    int j;

    printf("\nEnter the elements: \n");

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

        scanf("%d",&list[j]);

}

void swapList(int *m,int *n)

{

    int temp;

    temp = *m;

    *m = *n;

    *n = temp;

}

// Function for Bubble Sort

void bubblesort(int list[], int n)

{

    int i,j;

    for(i=0;i<(n-1);i++)

        for(j=0;j<(n-(i+1));j++)

            if(list[j] > list[j+1])

                swapList(&list[j],&list[j+1]);

}
```

```
// Showing the contents of the list
```

```
void printlist(int list[],int n)
{
    int j;
    for(j=0;j<n;j++)
        printf("%d\t",list[j]);
}

void main( )
{
    int list[MAX], num;
    clrscr( );
    printf("\n\n***** Enter the number of elements [Maximum 20] *****\n");
    scanf("%d",&num);
    readlist(list,num);
    printf("\n\nElements in the list before sorting are:\n");
    printlist(list,num);
    bubblesort(list,num);
    printf("\n\nElements in the list after sorting are:\n");
    printlist(list,num);
    getch( );
}
```

OUTPUT:

```
***** Enter the number of elements [Maximum 20] ***** 5
```

```
Enter the elements:
```

```
4  2  7  9  8
```

```
Elements in the list before sorting are:
```

```
4  2  7  9  8
```

```
Elements in the list after sorting are:
```

```
2  4  7  8  9
```

34. WRITE A C PROGRAM TO IMPLEMENT SELECTION SORT METHODS TO SORT A GIVEN LIST OF INTEGERS IN ASCENDING ORDER

```
#include<stdio.h>

#include<conio.h>

#define MAX 20

void readlist(int list[],int n)
{
    int j;
    printf("\nEnter the elements: \n");
    for(j=0;j<n;j++)
        scanf("%d",&list[j]);
}

void swapList(int *m,int *n)
{
    int temp;
    temp = *m;
    *m = *n;
    *n = temp;
}

void selectionsort(int list[], int n)
{
    int i,j,min;
    for(i=0;i<(n-1);i++)
    {
        min=i;
        for(j=i+1;j<n;j++)
            if(list[j]<list[min])
                min=j;
        if(min!=i)
            swapList(&list[i],&list[min]);
    }
}
```

```
// Showing the contents of the list
```

```
void printlist(int list[],int n)
{
    int j;
    for(j=0;j<n;j++)
        printf("%d\t",list[j]);
}

void main( )
{
    int list[MAX], num;
    clrscr( );
    printf("\n\n***** Enter the number of elements [Maximum 20] *****\n");
    scanf("%d",&num);
    readlist(list,num);
    printf("\n\nElements in the list before sorting are:\n");
    printlist(list,num);
    selectionsort(list,num);
    printf("\n\nElements in the list after sorting are:\n");
    printlist(list,num);
    getch( );
}
```

OUTPUT:

```
***** Enter the number of elements [Maximum 20] *****
```

```
5
```

```
Enter the elements:
```

```
9  1  4  8  3
```

```
Elements in the list before sorting are:
```

```
9  1  4  8  3
```

```
Elements in the list after sorting are:
```

```
1  3  4  8  9
```

35. WRITE A C PROGRAM TO IMPLEMENT INSERTION SORT METHODS TO SORT A GIVEN LIST OF INTEGERS IN ASCENDING ORDER

```
#include<stdio.h>

#include<conio.h>

#define MAX 20

void readlist(int list[],int n)
{
    int j;

    printf("\nEnter the elements: \n");

    for(j=0;j<n;j++)
        scanf("%d",&list[j]);
}

void insertionsort(int list[],intnum)
{
    int i,j,k;

    for(j=1;j<num;j++)
    {
        k=list[j];

        for(i=j-1;i>=0 && k<list[i];i--)
            list[i+1]=list[i];

        list[i+1]=k;
    }
}

// Showing the contents of the list

void printlist(int list[],int n)
{
    int j;

    for(j=0;j<n;j++)
        printf("%d\t",list[j]);
}
```

```
void main( )  
  
{  
  
    int list[MAX], num;  
  
    clrscr( );  
  
    printf("\n\n***** Enter the number of elements [Maximum 20] *****\n");  
  
    scanf("%d",&num);  
  
    readlist(list,num);  
  
    printf("\n\nElements in the list before sorting are:\n");  
  
    printlist(list,num);  
  
    insertionsort( list,num);  
  
    printf("\n\nElements in the list after sorting are:\n");  
  
    printlist(list,num);  
  
    getch( );  
  
}
```

OUTPUT:

```
***** Enter the number of elements [Maximum 20] *****
```

```
8
```

```
Enter the elements:
```

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

```
Elements in the list before sorting are:
```

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

```
Elements in the list after sorting are:
```

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


36. WRITE A C PROGRAM TO IMPLEMENT QUICK SORT METHODS TO SORT A GIVEN LIST OF INTEGERS IN ASCENDING ORDER

```
#include<stdio.h>

#include<conio.h>

#define MAX 20

void swap(int *m,int *n)

{

    int temp;

    temp = *m;

    *m = *n;

    *n = temp;

}

/*Function for Quick Sort*/

void quicksort(int list[],intm,int n)

{

    int key,i,j;

    if(m<n)

    {

        key = list[m];

        i = m;

        j = n;

        while(i <= j)

        {

            while((i <= n) && (list[i] <= key))

                i++;

            while((j >= m) && (list[j] > key))

                j--;

            if( i < j)

                swap(&list[i],&list[j]);

        }

    }

}
```

```
        swap(&list[m],&list[j]);

        quicksort(list,m,j-1);

        quicksort(list,j+1,n);

    }

}

/*Function to read the data*/

void readlist(int list[],int n)

{

    int j;

    printf("\n\nEnter the elements:\n");

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

        scanf("%d",&list[j]);

}

/*Function to print the data*/

void printlist(int list[],int n)

{

    int j;

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

        printf("%d\t",list[j]);

}

void main( )

{

    int list[MAX], num;

    clrscr( );

    printf("\n***** Enter the number of elements Maximum [20] *****\n");

    scanf("%d",&num);

    readlist(list,num);

    printf("\n\nElements in the list before sorting are:\n");

    printlist(list,num);
```

```
quicksort(list,0,num-1);

printf("\n\nElements in the list after sorting are:\n");

printlist(list,num);

getch( );

}
```

OUTPUT:

***** Enter the number of elements Maximum [10] *****

8

Enter the elements:

6 5 2 3 4 8 1 9

Elements in the list before sorting are:

6 5 2 3 4 8 1 9

Elements in the list after sorting are:

1 2 3 4 5 6 8 9

37. WRITE A C PROGRAM TO IMPLEMENT MERGE SORT METHODS TO SORT A GIVEN LIST OF INTEGERS IN ASCENDING ORDER

```
#include<stdio.h>

#include<conio.h>

#define MAX 20

void readlist(int list[],int n)

{

    int j;

    printf("\nEnter the elements: \n");

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

        scanf("%d",&list[j]);

}

/* Method to implement Merge Sort*/

void mergesort(int x[], int start, int end)

{

    int j = 0;

    const int size = end - start + 1;

    int mid = 0;

    int mrg1 = 0;

    int mrg2 = 0;

    int executing[size];

    if(end == start)

        return;

    mid = (start + end) / 2;

    mergesort(x, start, mid);

    mergesort(x, mid + 1, end);

    for(j = 0; j < size; j++)

        executing[j] = x[start + j];

    mrg1 = 0;
```

```
mrg2 = mid - start + 1;

for(j = 0; j < size; j++)
{
    if(mrg2 <= end - start)
        if(mrg1 <= mid - start)
            if(executing[mrg1] > executing[mrg2])
                x[j + start] = executing[mrg2++];
            else
                x[j + start] = executing[mrg1++];
        else
            x[j + start] = executing[mrg2++];
    else
        x[j + start] = executing[mrg1++];
}

}

// Showing the contents of the list

void printlist(int list[],int n)
{
    int j;
    for(j=0;j<n;j++)
        printf("%d\t",list[j]);
}

void main( )
{
    int list[MAX], num;

    clrscr();

    printf("\n\n***** Enter the number of elements [Maximum 20] *****\n");

    scanf("%d",&num);

    readlist(list,num);
```

```
printf("\n\nElements in the list before sorting are:\n");  
  
printlist(list,num);  
  
mergesort(list, 0, num-1);  
  
printf("\n\nElements in the list after sorting are:\n");  
  
printlist(list,num);  
  
getch( );  
  
}
```

OUTPUT:

```
***** Enter the number of elements [Maximum 20] *****
```

```
8
```

```
Enter the elements:
```

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

```
Elements in the list before sorting are:
```

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

```
Elements in the list after sorting are:
```

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

38a. WRITE A C PROGRAMS THAT USE NON RECURSIVE FUNCTIONS TO PERFORM LINEAR SEARCHING OPERATION FOR A KEY VALUE IN A GIVEN LIST OF INTEGERS

```
#include<stdio.h>

#include<conio.h>

#define MAX 20

void linearsearchnonrecursive(int list[ ],int num,int element);

void readlist(int list[ ],int num);

void printlist(int list[ ],int num);

void main( )

{

    int list[MAX], num, element;

    char c,y,n;

    clrscr( );

    printf("\n\n***** Enter the number of elements [Maximum 20] *****\n");

    scanf("%d",&num);

    readlist(list,num);

    printf("\nElements present in the list are:\n\n");

    printlist(list,num);

    while(1)

    {

        printf("\n\nDo you want to search element y/n : ");

        scanf(" %c",&c);

        switch(c)

        {
```

```
        case 'y':          printf("\n\nWhich Element you want to search: ");

                          scanf("%d",&element);

                          linearsearchnonrecursive(list,num,element);

                          break;

        case 'n':goto end;

    }

}

end:

getch( );

}

/* Non-Recursive method*/

void linearsearchnonrecursive(int list[ ],int num,int element)

{

    int j, f=0;

    for(j=0;j<num;j++)

        if( list[j] == element)

        {

            printf("\nThe element %d is present at position %d in list\n",element,j+1);

            f=1;

            break;

        }

    if(f==0)

        printf("\nThe element is %d is not present in the list\n",element);

}
```



```
void readlist(int list[ ],int num)

{

    int j;

    printf("\nEnter the elements:\n");

    for(j=0;j<num;j++)

        scanf("%d",&list[j]);

}

void printlist(int list[ ],int num)

{

    int j;

    for(j=0;j<num;j++)

        printf("%d\t",list[j]);

}
```

OUTPUT:

***** Enter the number of elements [Maximum 20] *****

8

Enter the elements:

2 5 6 9 8 7 4 3

Elements present in the list are:

2 5 6 9 8 7 4 3

Do you want to search element y/n : y

Which Element you want to search: 8

The element 8 is present at position 5 in list

ROLL NO:

Do you want to search element y/n : y

Which Element you want to search: 9

The element 9 is present at position 4 in list

Do you want to search element y/n : y

Which Element you want to search: 20

The element is 20 is not present in the list

Do you want to search element y/n : n

38b. WRITE A C PROGRAMS THAT USE RECURSIVE FUNCTIONS TO PERFORM LINEAR SEARCHING OPERATION FOR A KEY VALUE IN A GIVEN LIST OF INTEGERS

```
#include<stdio.h>

#include<conio.h>

#define MAX 20

void linearsearchrecursive(int list[ ],int num,int element);

void readlist(int list[ ],int num);

void printlist(int list[ ],int num);

void main( )

{

    int list[MAX], num, element;

    char c,y,n;

    clrscr( );

    printf("\n\n***** Enter the number of elements [Maximum 20] *****\n");

    scanf("%d",&num);

    readlist(list,num);

    printf("\nElements present in the list are:\n\n");

    printlist(list,num);

    while(1)

    {

        printf("\n\nDo you want to search element y/n : ");

        scanf(" %c",&c);

        switch(c)

        {

            case 'y':    printf("\n\nWhich Element you want to search: ");

                        scanf("%d",&element);

                        linearsearchrecursive(list,num,element);

                        break;
```

```
                case 'n' :goto end;
            }
        }
    end:
    getch( );
}
/* Recursive method*/
void linearsearchrecursive(int list[ ],int num,int element)
{
    int f = 0;
    if( list[num] == element)
    {
        printf("\nThe element %d is present at position %d in list\n",element,num+1);
        f=1;
    }
    else
    {
        if((num==0) && (f==0))
        {
            printf("The element %d is not found in the list.",element);
        }
        else
        {
            linearsearchrecursive(list,num-1,element);
        }
    }
}
```

```
void readlist(int list[ ],int num)
{
    int j;
    printf("\nEnter the elements:\n");
    for(j=0;j<num;j++)
        scanf("%d",&list[j]);
}

void printlist(int list[],int num)
{
    int j;
    for(j=0;j<num;j++)
        printf("%d\t",list[j]);
}
```

OUTPUT:

```
***** Enter the number of elements [Maximum 20] *****
```

```
8
```

```
Enter the elements:
```

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

```
Elements present in the list are:
```

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

```
Do you want to search element y/n : y
```

```
Which Element you want to search: 8
```

```
The element 8 is present at position 6 in list
```

```
Do you want to search element y/n : y
```

```
Which Element you want to search: 3
```

```
The element 3 is present at position 8 in list
```

```
Do you want to search element y/n : n
```

39a. WRITE A C PROGRAM THAT USE NON RECURSIVE FUNCTIONS TO PERFORM BINARY SEARCHING OPERATION FOR A KEY VALUE IN A GIVEN LIST OF INTEGERS

```
#include<stdio.h>

#include<conio.h>

#define MAX 20

/* Non-Recursive function*/

void bsearchnonrecursive(int list[ ],int num,int element)
{
    int low,high,mid, flag = 0;

    low = 0;

    high = num-1;

    while(low <= high)
    {
        mid = (low+high)/2;

        if( list[mid] == element)
        {
            printf("\nThe element %d is present at position %d in list\n",element,mid+1);

            flag =1;

            break;
        }

        else

            if(list[mid] < element)

                low = mid+1;

            else

                high = mid-1;

    }

    if( flag == 0)

        printf("\nThe element %d is not present in the list\n",element);
}
```

```
void readlist(int list[ ],int num)
{
    int i;
    printf("\nEnter the elements in sorted order :\n");
    for(i=0;i<num;i++)
        scanf("%d",&list[i]);
}
void printlist(int list[ ],int num)
{
    int i;
    for(i=0;i<num;i++)
        printf("%d\t",list[i]);
}
void main( )
{
    int list[MAX], num, element;
    char c,y,n;
    clrscr( );
    printf("\n\n\n***** Enter the number of elements [Maximum 20] *****\n");
    scanf("%d",&num);
    readlist(list,num);
    printf("\nElements present in the list are:\n\n");
    printlist(list,num);
    while(1)
    {
        printf("\n\nDo you want to search element y/n : ");
        scanf(" %c",&c);
        switch(c)
        {
            PROGRAMMING IN C AND DATA STRUCTURES
            KNREDDY
```

```
        case 'y':    printf("\n\nWhich Element you want to search: ");
                   scanf("%d",&element);
                   bsearchnonrecursive(list,num,element);
                   break;
        case 'n' :   goto end;
    }
}
end:
getch();
}
```

OUTPUT:

***** Enter the number of elements [Maximum 20] *****

8

Enter the elements in sorted order :

1 2 3 4 5 6 7 8

Elements present in the list are:

1 2 3 4 5 6 7 8

Do you want to search element y/n : y

Which Element you want to search: 8

The element 8 is present at position 8 in list

Do you want to search element y/n : y

Which Element you want to search: 9

The element 9 is not present in the list

Do you want to search element y/n : y

Which Element you want to search: 6

The element 6 is present at position 6 in list

Do you want to search element y/n : n

39b.WRITE A C PROGRAMS THAT USE RECURSIVE FUNCTIONS TO PERFORM BINARY SEARCHING OPERATION FOR A KEY VALUE IN A GIVEN LIST OF INTEGERS

```
#include<stdio.h>

#include<conio.h>

#define MAX 20

/* Recursive function*/

int bsearchrecursive(int list[ ],int low,int high,int element)
{
    int mid;
    if (low<=high)
    {
        mid=(low+high)/2;
        if (list[mid]==element)
            return mid;
        else if (element<list[mid])
            return bsearchrecursive(list,low,mid-1,element);
        else
            return bsearchrecursive(list,mid+1,high,element);
    }
    return -1;
}

void readlist(int list[ ],int num)
{
    int i;
    printf("\nEnter the elements in sorted order :\n");
    for(i=0;i<num;i++)
        scanf("%d",&list[i]);
}
```

```
void printlist(int list[ ],int num)
{
    int i;
    for(i=0;i<num;i++)
        printf("%d\t",list[i]);
}
void main( )
{
    int list[MAX], num, element,pos;
    char c,y,n;
    clrscr( );
    printf("\n\n\n***** Enter the number of elements [Maximum 20] *****\n");
    scanf("%d",&num);
    readlist(list,num);
    printf("\nElements present in the list are:\n\n");
    printlist(list,num);
    while(1)
    {
        printf("\n\nDo you want to search element y/n : ");
        scanf(" %c",&c);
        switch(c)
        {
            case 'y':    printf("\n\nWhich Element you want to search: ");
                        scanf("%d",&element);
                        pos=bsearchrecursive(list,1,num-1,element);
                        if(pos== -1)
                        {
                            printf("\n\nThe element %d is not present in the
                                list\n",element);
                        }
        }
    }
}
```

```

    }
    else
    {
        printf("\nThe element %d is present at position %d
in list\n",element,pos+1);
    }
    break;
    case 'n' :    goto end;
}
}
end:
getch( );
}

```

OUTPUT:

***** Enter the number of elements [Maximum 20] *****

8

Enter the elements in sorted order :

1 2 3 4 5 6 7 8

Elements present in the list are:

1 2 3 4 5 6 7 8

Do you want to search element y/n : y

Which Element you want to search: 8

The element 8 is present at position 8 in list

Do you want to search element y/n : y

Which Element you want to search: 9

The element 9 is not present in the list

Do you want to search element y/n : n

40. WRITE A PROGRAM TO TWO-WAY MERGE

```
#include<stdio.h>

#include<conio.h>

#include<string.h>

void main( )

{

    int a[100],b[100],c[100],m,n,i,j,nm,k;

    clrscr( );

    printf("\nenter number of elements of array a ");

    scanf("%d",&m);

    printf("\nenter number of elements of array b ");

    scanf("%d",&n);

    printf("\nEnter array a elements");

    for(i=0;i<m;i++)

        scanf("%d",&a[i]);

    printf("\nEnter array b elements");

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

        scanf("%d",&b[i]);

    if(a[m-1]<b[n-1])

        a[m]=b[n-1];

    else

        b[n]=a[m-1];

    i=0;

    j=0;

    nm=n+m;

    for(k=0;k<nm;k++)

    {

        if(a[i]<b[j])
```

```
{
    c[k]=a[i];
    i++;
}
else
{
    c[k]=b[j];
    j++;
}
}
printf("\n merged elements are\n");
for(i=0;i<nm;i++)
    printf("c[%d]=%d\t",i,c[i]);
getch();
}
```

OUTPUT:

enter number of elements of array a 5

enter number of elements of array b 3

Enter array a elements 1 3 4 6 9

Enter array b elements 5 7 8

merged elements are

c[0]=1 c[1]=3 c[2]=4 c[3]=5 c[4]=6 c[5]=7 c[6]=8 c[7]=9

41. WRITE A C PROGRAM TO EXCHANGE TWO NUMBERS USING POINTERS.

```
#include<stdio.h>

#include<conio.h>

void swap(int *x ,int *y);

void main( )
{
    int a,b;

    clrscr( );

    printf("enter any value for a:");

    scanf("%d",&a);

    printf("enter any value for b:");

    scanf("%d",&b);

    printf("\nvalues of a and b before exchange a=%d\tb=%d",a,b);

    swap(&a,&b);

    printf("\nvalues of a and b after exchange a=%d\tb=%d",a,b);

    getch( );

}

void swap(int *x, int *y)
{
    int temp;

    temp=*x;

    *x=*y;

    *y=temp;

}
```

OUTPUT:

enter any value for a:8

enter any value for b:16

values of a and b before exchange a=8 b=16

values of a and b after exchange a=16 b=8

42. WRITE A C PROGRAM TO READ A LIST OF STUDENTS INFORMATION AND PRINT THEM IN ASCENDING ORDER OF THEIR AVERAGE MARKS USING ARRAY OF STRUCTURES

```
#include<stdio.h>

#include<conio.h>

struct student

{

    char name[50],branch[5];

    int sub1,sub2,sub3;

    float avg;

}s[10];

void main( )

{

    int i,j,n;

    struct student temp;

    clrscr( );

    printf("\nEnter how many students:");

    scanf("%d",&n);

    for(i=1;i<=n;i++)

    {

        fflush(stdin);

        printf("\nEnter student %d Name:",i);

        gets(s[i].name);

        fflush(stdin);

        printf("\nEnter student %d Branch:",i);

        gets(s[i].branch);

        printf("\nEnter 3 subject Marks:");

        printf("\nsubject-1: ");

        scanf("%d",&s[i].sub1);

        printf("\nsubject-2: ");
```

```
scanf("%d",&s[i].sub2);

printf("\nsubject-3: ");

scanf("%d",&s[i].sub3);

}

for(i=1;i<=n;i++)

s[i].avg=(s[i].sub1+s[i].sub2+s[i].sub3)/3.0;

for(i=1;i<=n-1;i++)

{

    for(j=i+1;j<=n;j++)

    {

        if(s[i].avg>s[j].avg)

        {

            temp=s[i];

            s[i]=s[j];

            s[j]=temp;

        }

    }

}

printf("\nStudent Details As Per Their Averege Marks Are:");

for(i=1;i<=n;i++)

printf("\n%s\t%s\t%.2f",s[i].name,s[i].branch,s[i].avg);

getch( );

}
```


OUTPUT:

Enter how many students:3

Enter student 1 Name:babu

Enter student 1 Branch:ece

Enter 3 subject Marks:

subject-1: 70

subject-2: 89

subject-3: 67

Enter student 2 Name:knr

Enter student 2 Branch:cse

Enter 3 subject Marks:

subject-1: 79

subject-2: 89

subject-3: 87

Enter student 3 Name:raj

Enter student 3 Branch:civil

Enter 3 subject Marks:

subject-1: 67

subject-2: 78

subject-3: 89

Student Details As Per Their Average Marks Are:

babu	cse	75.33
------	-----	-------

raj	civil	78.00
-----	-------	-------

knr	cse	85.00
-----	-----	-------

43. Write a C program to read student records into a file. Record consists of rollno, name and marks of a student in six subjects and class. Class field is empty initially. Compute the class of a student. The calculation of the class is as per JNTUA rules. Write the first class, second class, third class and failed students lists separately to another file.

/*Write a program to read student records into a file. Record consists of rollno, name and marks of a student in six subjects and class. Class field is empty initially. Compute the class of a student. The calculation of the class is as per JNTUA rules. Write the first class, second class, third class and failed students lists separately to another file.*/

```
#include<stdio.h>

#include<conio.h>

struct student
{
    char roll[10],name[50],branch[5],*class;
    int sub1,sub2,sub3,sub4,sub5,sub6;
    float avg;
}s[10];

void main( )
{
    FILE *fpd,*fp1,*fp2,*fp3,*fpf;
    int i,j,n;
    struct student temp;
    clrscr ( );
    printf("\nEnter how many students:");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        fflush(stdin);
        printf("\nEnter student %d Name:",i);
        gets(s[i].name);
        fflush(stdin);
        printf("\nEnter roll number");
```

```
    gets(s[i].roll);

    printf("\nEnter student %d Branch:",i);

    gets(s[i].branch);

    printf("\nEnter 6 subject Marks:");

    printf("\nsubject-1: ");

    scanf("%d",&s[i].sub1);

    printf("\nsubject-2: ");

    scanf("%d",&s[i].sub2);

    printf("\nsubject-3: ");

    scanf("%d",&s[i].sub3);

    printf("\nsubject-4: ");

    scanf("%d",&s[i].sub4);

    printf("\nsubject-5: ");

    scanf("%d",&s[i].sub5);

    printf("\nsubject-6: ");

    scanf("%d",&s[i].sub6);

    s[i].avg=(s[i].sub1+s[i].sub2+s[i].sub3+s[i].sub4+s[i].sub5+s[i].sub6)/6.0;

    if(s[i].avg >= 70)

        s[i].class="DIST";

    else if(s[i].avg<70&& s[i].avg>=60)

        s[i].class="FIRST";

    else if(s[i].avg<60&& s[i].avg>=50)

        s[i].class="SECOND";

    else if(s[i].avg<50&& s[i].avg>=40)

        s[i].class="THIRD";

    else

        s[i].class="FAIL";

}
```

```
fpd=fopen("dist.txt","w");
```

```
fp1=fopen("first.txt","w");

fp2=fopen("second.txt","w");

fp3=fopen("third.txt","w");

fpf=fopen("fail.txt","w");

fprintf(fpd,"NAME\tROLL\tBRANCH\tCLASS\n");

fprintf(fp1,"NAME\tROLL\tBRANCH\tCLASS\n");

fprintf(fp2,"NAME\tROLL\tBRANCH\tCLASS\n");

fprintf(fp3,"NAME\tROLL\tBRANCH\tCLASS\n");

fprintf(fpf,"NAME\tROLL\tBRANCH\tCLASS\n");

for(i=1;i<=n;i++)
{
    if(s[i].class=="DIST")

        fprintf(fpd,"%s\t%s\t%s\t%s\n",s[i].name,s[i].roll,s[i].branch,s[i].class);

    else if(s[i].class=="FIRST")

        fprintf(fp1,"%s\t%s\t%s\t%s\n",s[i].name,s[i].roll,s[i].branch,s[i].class);

    else if(s[i].class=="SECOND")

        fprintf(fp2,"%s\t%s\t%s\t%s\n",s[i].name,s[i].roll,s[i].branch,s[i].class);

    else if(s[i].class=="THIRD")

        fprintf(fp3,"%s\t%s\t%s\t%s\n",s[i].name,s[i].roll,s[i].branch,s[i].class);

    else

        fprintf(fpf,"%s\t%s\t%s\t%s\n",s[i].name,s[i].roll,s[i].branch,s[i].class);

}

fclose(fpd);

fclose(fp1);

fclose(fp2);

fclose(fp3);

fclose(fpf);

getch( );

}
```

44. WRITE A C PROGRAM TO MERGE TWO FILES INTO A THIRD FILE

```
#include<stdio.h>

#include<conio.h>

void main( )
{
    char ch;

    FILE *f1,*f2,*f3;

    clrscr();

    f1=fopen("demo1.txt","r");

    f3=fopen("result.txt","w");

    if(f1==NULL)
    {
        printf("\nFile Opening Error");
    }

    while((ch=fgetc(f1))!=EOF)

    fputc(ch,f3);

    fclose(f1);

    fclose(f3);

    f2=fopen("demo2.txt","r");

    f3=fopen("result.txt","a");

    if(f2==NULL)
    {
        printf("\nFile Opening Error");
    }

    while((ch=fgetc(f2))!=EOF)

    fputc(ch,f3);

    fclose(f2);

    fclose(f3);

    f3=fopen("result.txt","r");
```

```
    if(f3==NULL)
    {
        printf("\nFile Opening Error");
    }
    printf("\nMERGING CONTENTS ARE:\n");
    while(!feof(f3))
        printf("%c",getc(f3));
    fclose(f3);
    getch( );
}
```

NOTE: For the above program first we need to create two files with the name demo1.txt and demo2.txt with some content in the two files.

45. WRITE A C PROGRAMS THAT IMPLEMENT STACK (ITS OPERATIONS) USING ARRAYS

```
#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

int top=-1,*stackarr,item,size;

void push( );

void pop( );

void display( );

void main( )

{

    int choice;

    int n;

    clrscr( );

    printf("\nEnter number of elements to be entered in an array");

    scanf("%d",&n);

    stackarr=(int*)calloc(n,sizeof(int));

    size=n;

    while(1)

    {

        printf("\n=====");

        printf("\n\t\t MENU ");

        printf("\n=====");

        printf("\n[1] Using Push Function");

        printf("\n[2] Using Pop Function");

        printf("\n[3] Elements present in Stack");

        printf("\n[4] Exit\n");

        printf("\n\tEnter your choice: ");

        scanf(" %d",&choice);
```

```
switch(choice)
{

    case 1: push( );
            break;

    case 2: pop( );
            break;

    case 3: display();
            break;

    case 4: goto end;

    default: printf("\nYour choice is invalid.\n");

}

}

end:

getch( );

}

/*Implementing the push() function. */

void push()
{
    if(top==size-1)
        printf("*** stack is full ***\n");
    else
    {
        printf("enter the item to be pushed into the stack:");
        scanf("%d",&item);
        stackarr[++top]=item;
    }
    printf("top=%d\n",top);
}
```



```
/*Implementing the pop() function. */

void pop()

{

    if(top== -1)

        printf("*** stack is empty ***\n");

    else

    { item=stackarr[top];

      top--;

      printf("the deleted item from stack is %d\n",item);

    }

    printf("top=%d\n",top);

}

/*Implementing display() function. */

void display()

{

    int k;

    printf("\n\tElements present in the stack are:\n\t");

    for(k=0;k<=top;k++)

        printf("%d\t",stackarr[k]);

}
```

46. WRITE A C PROGRAMS THAT IMPLEMENT QUEUE (ITS OPERATIONS) USING ARRAYS

```
#include<stdio.h>

#include<conio.h>

#define max 50

int q[max],front=-1,rear=-1;

int size;

void insert();

void delet();

void display();

void front_of_queue();

void main( )

{

    int choice;

    printf("\n enter the size of queue");

    scanf("%d",&size);

    clrscr( );

    while(1)

    {

        printf("\n=====");

        printf("\n\t\t MENU ");

        printf("\n=====");

        printf("\n[1] Using insert Function");

        printf("\n[2] Using delete Function");

        printf("\n[3] Elements present in Queue");

        printf("\n[4] front of queue\n");

        printf("\n[5] Exit\n");

        printf("\n\tEnter your choice: ");

        scanf(" %d",&choice);
```

```
        switch(choice)
        {
            case 1: insert();
                    break;
            case 2: delet();
                    break;
            case 3: display();
                    break;
            case 4: front_of_queue();
                    break;
            case 5: goto end;
            default: printf("\nYour choice is invalid.\n");
                    break;
        }
    }
    end:
    getch();
}

void insert()
{
    int x;
    if(rear>=size-1)
        printf("\n queue overflow");
    else
    {
        printf("\n enter the element");
        scanf("%d",&x);
        if(rear==size-1)
            rear=front=0;
        else
            rear++;
        q[rear]=x;
    }
}
```

```
    }  
}  
void delet( )  
{  
    if(front<0)  
        printf("queue underflow");  
    else  
    {  
        printf("the element deleted is %d\t",q[front]);  
        if(front==rear)  
            front=rear=-1;  
        else  
            front++;  
    }  
}  
void display( )  
{  
    int i;  
    if(front<0)  
        printf("queue underflow");  
    else  
    {  
        for(i=front;i<=rear;i++)  
            printf("%d\n",q[i]);  
    }  
}  
void front_of_queue( )  
{  
    if(front<0)  
        printf("queue underflow");  
    else  
        printf("the front of queue is %d",q[front]);  
}
```

47. WRITE A C PROGRAM THAT USES STACK OPERATIONS TO PERFORM THE FOLLOWING:

I) CONVERTING INFIX EXPRESSION INTO POSTFIX EXPRESSION

II) EVALUATING THE POSTFIX EXPRESSION

```
#include<stdio.h>

#include<conio.h>

#include<ctype.h>

#include<stdlib.h>

#include<string.h>

int st[100];

int st_top=-1;

int cal(char post[]);

void in_post(char in[]);

void push_item(int it);

int pop_item();

int st_ISP(char t);

int st_ICP(char t);

/*main function*/

void main( )

{

    char in[100],post[100];

    clrscr()

    printf("\n\tEnter the Infix Expression: ");

    gets(in);

    in_post(in);

    getch( );

}

/*end main*/
```

```
void push_item(int it)
{
    if(st_top==99)
    {
        printf("\n\n\t*STACK is Full*");
        getch( );
        exit(1);
    }
    st[++st_top]=it;
}

int pop_item()
{
    int it;
    if(st_top== -1)
    {
        getch( );
    }
    return(st[st_top--]);
}

/*Function for converting an infix expression to a postfix expression. */
void in_post(char in[])
{
    int x=0,y=0,z,result=0;
    char a,c, post[100];
    char t;
    push_item('\0');
    t=in[x];
    while(t!='\0')
    {
```

```
if(isalnum(t))

/*For checking whether the value in t is an alphabet or number. */

{

    post[y]=t;

    y++;

}

else if(t=='(')

{

    push_item('(');

}

else if(t==')')

{

    while(st[st_top]!='(')

    {

        c=pop_item();

        post[y]=c;

        y++;

    }

    c=pop_item();

}

else

{

    while(st_ISP(st[st_top])>=st_ICP(t))

    {

        c=pop_item();

        post[y]=c;

        y++;

    }

    push_item(t);

}
```

```
x++;  
t=in[x];  
}  
while(st_top!=-1)  
{  
    c=pop_item();  
    post[y]=c;  
    y++;  
}  
printf("\n\tThe Postfix Expression is:");  
for(z=0;z<y;z++)  
    printf("%c",post[z]);  
printf("\n\nDo you want to evaluate the Result of Postfix Expression?(Y/N):");  
scanf("%c",&a);  
if(a=='y' || a=='Y')  
{  
    result=cal(post);  
    printf("\n\n\tResult is: %d\n",result);  
    getch( );  
}  
else if(a=='n' || a=='N')  
{  
    exit(0);  
}  
}
```



```
/*Determining priority of inside elements*/

int st_ISP(char t)
{
    switch(t)
    {
        case '(':return (10);
        case ')':return (9);
        case '+':return (7);
        case '-':return (7);
        case '*':return (8);
        case '/':return (8);
        case '\0':return (0);
        default: printf("Expression is invalid.");
        break;
    }
    return 0;
}
```

```
/*Determining priority of approaching elements*/
```

```
int st_ICP(char t)
{
    switch(t)
    {
        case '(':return (10);
        case ')':return (9);
        case '+':return (7);
        case '-':return (7);
        case '*':return (8);
        case '/':return (8);
        case '\0':return (0);
```

```
        default: printf("Expression is invalid.");
        break;
    }
    return 0;
}
/*Evaluating the result of postfix expression*/
int cal(char post[])
{
    int m,n,x,y,j=0,len;
    float a;
    len=strlen(post);
    while(j<len)
    {
        if(isalpha(post[j]))
        {
            printf("\n enter the value of %c: ",post[j]);
            scanf("%f",&a);
            push_item(a);
        }
        else
        {
            m=pop_item( );
            n=pop_item();
            switch(post[j])
            {
                case '+':x=n+m;
                    break;
                case '-':x=n-m;
                    break;
```

```
                case '*':x=n*m;
                    break;
                case '/':x=n/m;
                    break;
            }
            push_item(x);
        }
        j++;
    }
    if(st_top>0)
    {
        printf("Number of Operands are more than Operators.");
        exit(0);
    }
    else
    {
        y=pop_item();
        return (y);
    }
    return 0;
}
```