/* 16. Design and develop a function matchany(s1,s2) which returns the first location in the string s1 where any character from the string s2 occurs, or -1 if s1 contains no character from s2.Do not use the standard library function which does a similar job.Invoke the function matchany(s1,s2) from the main for different strings and print both the strings and return value from the function matchany(s1,s2). */

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
int matchany(char str[],char str2[]);
void main()
{
        char str1[50],str2[50];
        int s;
        clrscr();
        printf("Enter string 1");
        scanf("%s",&str1);
        printf("Enter string 2");
        scanf("%s",&str2);
        s=matchany(str1,str2);
        if(s==-1)
                printf("No match occured");
        else
                printf("The location where the first match occured is %d",s);
        getch();
}
int matchany(char str1[],char str2[])
{
        int i,j;
        for(i=0;i<strlen(str2);i++)
        {
                for(j=0;j<strlen(str1);j++)
                {
                        if(str2[i]==str1[j])
                        {
                         return j+1;
                        }
                }
        }
        return -1;
}
```

9.Design,develop and execute a program in C to calculate the approximate value of exp(0.5) using the taylor series expansion for the exponential function. Use the terms in the expansion until the last term is less than the machine epsilon defines as FLT_EPSILON in the header file <float.h>. Also print the value returned by the Mathematical function exp( ).

```
#include<stdio.h>
#include<conio.h>
#include<float.h>
#include<math.h>
float factorial(long int n)
{
        long int fact=1,tmp=1;
        while(tmp<=n)
        {
                fact=fact*tmp;
                tmp++;
        }
        return fact;
 }
void main()
 {
        float x=0,temp=0,result=0;
        int i=0;
        clrscr();
        printf("Enter x:");
        scanf("%f",&x);
        printf("\nx=%f",x);
        do
        {
                temp=(pow(x,i)/factorial(i));
                result=result+temp;
                i++;
        }while(temp>FLT_EPSILON);
        printf("\n e^%f=%f",x,result);
        printf("\nTheoretical vlaue=%f",exp(x));
        getch();
 }
```

1. Design, develop and execute a program in C to find and output all the roots of a given quadratic equation, for non-zero coefficients.

```
#include<stdio.h>
#include<conio.h>
#include<process.h>
#include<math.h>
main()
{
  float a,b,c,disc,x1,x2;
  clrscr();
  printf("Enter the values for a ,b and c:");
  scanf("%f %f %f",&a,&b,&c);
  if(a*b*c==0)
  {
        printf("The values are not allowed");
        getch();
        exit(0);
  }
  disc=b*b-4*a*c;
  if(disc==0)
  {
        x1=x2=(-b)/(2*a);
        printf("The equal and real roots are:");
        printf("x1=x2=%f",x1);
  }
  else if(disc>0)
  {
        x1=(-b+sqrt(disc))/(2*a);
        x2=(-b-sqrt(disc))/(2*a);
        printf("The real and distinct roots are:\n");
        printf("x1=%f\tx2=%f",x1,x2);
  }
  else
  {
        x1=-b/(2*a);
        x2=sqrt(fabs(disc))/(2*a);
        printf("The complex roots are:\n");
```

```
        printf("x1=%f+i%f\t",x1,x2);
        printf("x1=%f-i%f\t",x1,x2);
  }
  getch();
}
```

13. Design and develop a function isprime (x) that accepts an integer argument and returns 1 if the argument is prime and 0 otherwise. The function must use plain division checking approach to determine if a given number is prime. Invoke this function from the main with different values obtained from the user and print appropriate messages.

```
#include<stdio.h>
#include<conio.h>
int x,flag=1,i,r;
int isprime(int x)
{
        for(i=2;i<=x/2;i++)
        {
                if(x%i==0)
                {
                        flag=0;
                }

        }
        return flag;
}
void main()
{
clrscr();
printf("Enter a number greater than 1 \n");
scanf("%d",&x);
r=isprime(x);
if(r==0)
printf("The given number %d is not prime\n",x);
else
printf("The given number %d is prime\n",x);
getch();
}
```

10. Design, develop and execute a program in C to read two matrices A (M x N) and B (P x Q) and compute the product of A and B if the matrices are compatible for multiplication. The program must print the input matrices and the resultant matrix with suitable headings and format if the matrices are compatible for multiplication, otherwise the program must print a suitable message. (For the purpose of demonstration, the array sizes M, N, P, and Q can all be less than or equal to 3.

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

int i,j,k,m,n,p,q,a[10][10],b[10][10],c[10][10];

void input(int a[10][10],int m,int n)
{
for (i=0;i<m;i++)
for(j=0;j<n;j++)
scanf("%d",&a[i][j]);
}

void multiply()
{
        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];
        }
}
void output(int a[10][10],int m,int n)
{
        for(i=0;i<m;i++)
        for(j=0;j<n;j++)
        {
                printf("%d\t" ,a[i][j]);
        }
```

```
    printf("\n");

}



void main()
{
clrscr();
printf("ebter the order of matrix A\n");
scanf("%d %d",&m,&n);
printf("enter the order of matrix B\n");
scanf("%d %d",&p,&q);
if(n!=p)
{
printf("Multification is not possible");
getch();
exit(0);
}
printf("Enter the elements of matrix A\n");
input(a,m,n);
printf("Enter the elements of matrix B\n");
input(b,p,q);
multiply();
printf("the matrix A is :\n");
output(a,m,n);
printf("the matrix B is :\n");
output(b,p,q);
printf("the output matrix:\n");
output(c,m,q);
getch();
}
```

14. Design, develop and execute a parallel program in C to determine and print the prime numbers which are less than 100 making use of algorithm of the Sieve of Eratosthenes.

```
#include<stdio.h>
#include<omp.h>
#define  size 100
#define TRUE 1
#define FALSE 0
main()
{
        int a[size],N;
        int i,k,threads,pcount;
        int found;
        printf("Enter the number of threads \n");
        scanf("%d",&threads);


        omp_set_num_threads(threads);
#pragma omp parallel for
        for(i=2;i<100;i++)
        {
                a[i]=TRUE;
        }
        k=2;
        while(k*k<100)
        {
#pragma omp parallel for
                for(i=k*k;i<=100;i+=k)
```

```
            {
                a[i]=FALSE;
            }

            k++;
        }
        pcount=0;
        printf("\n list of prime nos \n");
        for(i=2;i<=100;i++)
        {
            if(a[i]==TRUE)
            {
                pcount++;
                printf("%d\n",i);
            }
        }
        printf("\n there are %d primes between 0 and %d\n",pcount,100);
        getch();
}
```

5. Design, develop and execute a program in C to copy its input to its output, replacing each string of one or more blanks by a single blank.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
        char input[100],output[100],c;
        int i=0,j=0;
        clrscr();
        printf("enter the string\n");
        gets(input);
        for(i=0;i<strlen(input);i++)
        {
            if(input[i]==' ' && input[i+1]!=' ')
            output[j++]=input[i];
            else if(input[i]!=' ')
            output[j++]=input[i];
```

```
        }
        output[j]='\0';
        printf("\n");
        puts(input);
        printf("\n");
        puts(output);
        getch();
}
```

4. Design, develop and execute a program in C to evaluate the given polynomial f(x) = a4x4 + a3x3 + a2x2 + a1x + a0 for given value of x and the coefficients using Horner's method.

```
#include<stdio.h>
#include<conio.h>
void main()
{
        float a[100],sum=0,x;
        int n,i;
        clrscr();
        printf("Enter the degree of the polynomial:");
        scanf("%d",&n);
        printf("\n Enter %d the coefficients into array:",n);
        for(i=n;i>=0;i--)
        {
                scanf("%f",&a[i]);
        }
```

```
        printf("Enter the value of x:\n");
        scanf("%f",&x);
        for(i=n;i>0;i--)
        {       sum=(sum+a[i])*x;
        }
        sum=sum+a[0];
        printf("\n The value of f(%.2f)=%.2f",x,sum);
        getch();
}
```

11. Design, develop and execute a parallel program in C to add, elementwise, two one-dimensional arrays A and B of N integer elements and store the result in another one-dimensional array C of N integer elements.

```
#include<stdio.h>
#include<omp.h>
#include<conio.h>
#define N 100
#define NUM_THREADS 4
main()
```

```
{
        int a[N],b[N],c[N];
        int n,r,i;
        printf("Enter the number of elements \n");
        scanf("%d",&n);
        printf("Enter the elements of array A \n");
        for(i=0;i<n;i++)
                scanf("%d",&a[i]);
                printf("Enter the elements of array B \n");
                for(i=0;i<n;i++)
                        scanf("%d",&b[i]);
                omp_set_num_threads(NUM_THREADS);
                #pragma omp parallel for
                for(i=0;i<n;i++)
                {
                        c[i]=a[i]+b[i];
                }
                printf(" The result in array C is \n");
                for(i=0;i<n;i++)
                        printf("%d",c[i]);
                getch();
}
```

7. Design, develop and execute a program in C to input N integer numbers into a single dimensional array, sort them in ascending order using bubble sort technique and print both the given array and the sorted array with suitable headings.

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int i,j,n,temp,a[100];
        clrscr();
        printf("Enter the value for n:");
        scanf("%d",&n);
        printf("Enter the array elements :");
        for(i=0;i<n;i++)
        {
                scanf("%d", &a[i]);
        }
        printf("the unsorted array is :\n");
        for(i=0;i<n;i++)
        {
                printf("%d\t",a[i]);
        }
        for(i=1;i<n;i++)
        {

                for(j=0;j<(n-i);j++)
                {
                        if(a[j]>a[j+1])
                        {
                        temp=a[j];
                        a[j]=a[j+1];
                        a[j+1]=temp;
                }
        }
}
printf("the sorted array is:\n");
for(i=0;i<n;i++)
printf("%d\t", a[i]);
getch();
}
```

6.Design, develop and execute a program in C to input N integer numbers in ascending order into a single dimension array, and then to performa binary search for a given key integer number and report success or failure in the form of a suitable message.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[100],key,i,n,mid,last,first,flag=0;
clrscr();
printf("Enter the value of n:");
scanf("%d",&n);
printf("Enter the elements:");
{
for(i=0;i<n;i++)
scanf("%d",&a[i]);
}
first=0;
last=n-1;
printf("Enter the key:");
scanf("%d",&key);
while(first<=last)
{
mid=(last+first)/2;
if(key==a[mid])
{
        printf("Search successful\n");
        printf("key found at location %d",mid+1);
        flag=1;
        break;
}
else if(key>a[mid])
{
        first=mid+1;
}
else
{
        last=mid-1;
}
}
if(flag==0)
{
        printf("search unsuccessful");
}
getch();
}
```

2.Design, develop and execute a program in C Euclid's algorithm to find the GCD and LCM of two integers and to output the results along with the given integers.

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int num1,num2,temp1,temp2,rem,gcd,lcm;
        clrscr();
        printf("Enter two integers num1 and num2 \n");
        scanf("%d %d",&num1,&num2);
        temp1=num1;
        temp2=num2;
        while(num2!=0)
        {
                rem=num1%num2;
                num1=num2;
                num2=rem;
        }
        gcd=num1;
        lcm=(temp1*temp2)/gcd;
        printf("the gcd of %d and %d is =%d \n",temp1,temp2,gcd);
        printf("the lcm of %d and %d is =%d",temp1,temp2,lcm);
        getch();
}
```

12.Design and develop a function rightrot(x,n) in C that returns the value of the integer x rotated to the right by n bit position as an unsigned integer.Invoke the function from the main with different values for x and n and print the results with suitable headings.

```
#include<stdio.h>
#include<conio.h>
unsigned int rightrot(unsigned x, unsigned n)
{
unsigned int y,z,a;
a=x>>n;
z=x<<(16-n);
y=a|z;
return (y);
}
void main()
{
unsigned x;
unsigned r,n;
clrscr();
printf("Enter the value of x\n");
scanf("%u",&x);
printf("Enter the value of n\n");
scanf("%u",&n);
r=rightrot(x,n);
printf("x=%u n=%u:r=%u\n",x,n,r);
getch();
}
```

15.Design and develop a function reverse(s) in C to reverse the srings in place.Invoke this function from the main for different strings and print the orginal and reversed strings.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
        char str[50],temp;
        int i,len;
        clrscr();
        printf("Enter a string:");
        scanf("%s", &str);
        len=strlen(str)-1;
        for (i=0;i<strlen(str)/2;i++)
        {
                temp=str[i];
                str[i]=str[len];
                str[len--]=temp;
        }
        printf("%s",str);
        getch();
}
```

3.Design.develop and execute a program in C to reverse a given four digit integer number and check whether it is palindrome or not.Output the given number with suitable message.

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

void main()
{
        int temp,num,rev=0,rem;
        clrscr();
        printf("Enter four digit integers:\n");
        scanf("%d",&num);
        temp=num;
        do
        {
                rem=num%10;
                rev=rev*10+rem;
                num=num/10;
        }
        while(num!=0);
        if(temp==rev)
        {
                printf("the number %d is palindrome \n",temp);
        }
        else
        {
                printf("the number %d is not palindrome", temp);
        }
        getch();
}
```

8.Design,develop and execute a program in C to compute and print the word length on the host machine.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,x;
clrscr();
x=sizeof(a);
printf("%d\n",sizeof(a));
if(x==2)
{
printf("Word length of host machine is 16 bits");
}
else
{
printf("Its a 32 bit machine");
}
getch();
}
```

Computer programming laboratory

Total Hrs. : 42 Exam Marks : 50

PART – A

1. Design, develop and execute a program in C to find and output all the roots of a given quadratic equation, for non-zero coefficients.
2. Design, develop and execute a program in C to implement Euclid's algorithm to find the GCD and LCM of two integers and to output the results along with the given integers.
3. Design, develop and execute a program in C to reverse a given four digit integer number and check whether it is a palindrome or not. Output the given number with suitable message.
4. Design, develop and execute a program in C to evaluate the given polynomial f(x) = a4x4 + a3x3 + a2x2 + a1x + a0 for given value of x and the coefficients using Horner's method.
5. Design, develop and execute a program in C to copy its input to its output, replacing each string of one or more blanks by a single blank.
6. Design, develop and execute a program in C to input N integer numbers in ascending order into a single dimensional array and perform a binary search for a given key integer number and report success or failure in the form of a suitable message.
7. Design, develop and execute a program in C to input N integer numbers into a single dimensional array, sort them in ascending order using bubble sort technique and print both the given array and the sorted array with suitable headings.
8. Design, develop and execute a program in C to compute and print the word length on the host machine.

PART – B

9. Design, develop and execute a program in C to calculate the approximate value of exp(0.5) using the Taylor Series expansion for the exponential function. Use the terms in the expansion until the last term is less than the machine epsilon defined FLT_EPSILON in the header file <float.h>. Also print the value returned by the Mathematical function exp( ).
10. Design, develop and execute a program in C to read two matrices A (M x N) and B (P x Q) and compute the product of A and B if the matrices are compatible for multiplication. The program must print the input matrices and

the resultant matrix with suitable headings and format if the matrices are compatible for multiplication, otherwise the program must print a suitable message. (For the purpose of demonstration, the array sizes M, N, P, and Q can all be less than or equal to 3)
11. Design, develop and execute a parallel program in C to add, elementwise, two one-dimensional arrays A and B of N integer elements and store the result in another one-dimensional array C of N integer elements.

12. Design and develop a function rightrot (x, n) in C that returns the value of the integer x rotated to the right by n bit positions as an unsigned integer. Invoke the function from the main with different values for x and n and print the results with suitable headings.
13. Design and develop a function isprime (x) that accepts an integer argument and returns 1 if the argument is prime and 0 otherwise. The function must use plain division checking approach to determine if a given number is prime. Invoke this function from the main with different values obtained from the user and print appropriate messages.
14. Design, develop and execute a parallel program in C to determine and print the prime numbers which are less than 100 making use of algorithm of the Sieve of Eratosthenes.
15. Design and develop a function reverses (s) in C to reverse the string s in place. Invoke this function from the main for different strings and print the original and reversed strings.
16. Design and develop a function match any (s1,s2) which returns the first location in the string s1 where any character from the string s2 occurs, or – 1 if s1 contains no character from s2. Do not use the standard library function which does a similar job! Invoke the function match any (s1. s2) from the main for different strings and print both the strings and the return value from the function match any (s1,s2).

Note: In the practical examination, the student has to answer two questions. One question from Part A and one question from Part B will be selected by the student by lots. All the questions listed in the syllabus have to be included in the lots. The change of question (Part A only / Part B only / Both Part A & Part B) has to be considered, provided the request is made for the same, within half an hour from the start of the examination. The allotment of marks is as detailed below:

Sl. No.	Activity	Max. Marks
1.	Procedure Part A Writing program & procedure for the assigned problems along with algorithms / flowchart Part B	5* 5*
2.	Conduction Execution of the program and Part A	10