Name of Course	: **CBCS B.Sc. (H) Mathematics**
Unique Paper Code	: **32357503**
Name of Paper	: **DSE-I C++ Programming for Mathematics**
Semester	: **V**
Duration	: **3 hours**
Maximum Marks	: **75 Marks**

Attempt any four questions. All questions carry equal marks.

1. Write C++ code to implement the mathematical *mod* function (name it *myMod*) without using the inbuilt operator % or any inbuilt function. The function must take two integers as inputs and must return the positive remainder when the large number is divided by the small number (ignoring the sign). Further, use this function *myMod* to find the gcd of two integers using the *While* loop **(8+10.75 Marks).**

2.

a. Describe passing arguments through call by value and call by reference to a function. Discuss the difference through examples **(6 Marks).**

b. Write a C++ program that takes a day, a month, and a year as input and calculate a date 25 days after the input date. Print this new date in the format dd/mm/yyyy **(12.75 Marks).**

3. Write a program to read 10 integers from the user and store them in an array named **intlist.** Pass this array to the following procedures:

a. **COUNT** that prints the total number of positive integers, negative integers and zeros in intlist **(8 Marks).**

b. **SORTLIST** that sorts the inlist elements by storing all positive integers, followed by zeros and finally all negative integers. For example if intlist originally consists of 1, -2, 5, 0, -6, 7, 0, -4, 0, 0 then after calling SORTLIST, intlist becomes -2, -6, -4, 0, 0, 0, 0, 1, 5, 7 **(10.75 Marks).**

4. Create a *Matrix* class to represent an element of the ring

$$M_2(\mathbb{R}) = \left\{ \begin{bmatrix} a & b \\ c & d \end{bmatrix} \middle| \, a, b, c, d \in \mathbb{R} \right\}$$

with respect to the usual matrix addition and multiplication.

The class should have the following features :

i. The *Matrix* class should have four private data elements:

a, b, c and *d* (of double type) to represent the 2×2 real matrix *A:*

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

P.T.O.

ii. The class should include the following constructors :

a. A default constructor to represent the 2×2 zero matrix.

b. A four argument constructor.

iii. Include get methods to inspect the values held in *a, b, c* and *d*.

iv. Include set methods to change the values held in *a, b, c* and *d*.

v. Include the operator + to add two *Matrix* objects.

vi. Include the operator * to multiply two *Matrix* objects.

vii. Include an operator << for printing a *Matrix* object in the form :

$$\begin{array}{cc} a & b \\ c & d \end{array}$$

Create any two objects of above class matrices and perform the addition(+) and multiplication (*) of these objects. Further display these two objects in matrix form **(18.75 Marks).**

5. Write a program to input a sentence from the user and pass it to the following two procedures:
   a. **COUNTCASE** that returns the total number of Uppercase and Lowercase alphabets in it and a procedure **(9.75 Marks).**
   b. **CHECKSPECIAL** that checks if special characters ! and ? are present and return TRUE/FALSE for both these characters **(9 Marks).**

6.
   a. Write a program in C++ to find 100! using a procedure with the help of GMP package **(9 Marks).**
   b. Write a C++ program to generate 50 numbers randomly from the set

      $$\{1, 2, \ldots, 10\}$$

      and write them to a file named **random.txt**. Then the same program must find the average of the 50 numbers from the file **random.txt** and write the average to a file named **average.txt** **(9.75 Marks).**

DSE-I C++ Programming for Mathematics. Answer Key.

Q1.

```
#include <iostream>
#include<cmath>
using namespace std;
int myMod(int, int);
int main(){
   int a, b, rem;
   cout<<"Enter the first integer "<<endl;
   cin>>a;
   cout<<"Enter the second integer "<<endl;
   cin>>b;
   a = abs(a);
   b = abs(b);
   if(a==0 && b!=0){
       cout<<"GCD is "<<b<<endl;
       return 0;
   }
   if(a!=0 && b==0){
       cout<<"GCD is "<<a<<endl;
       return 0;
   }
   rem = myMod(a, b);
   while(rem!=0){
       a = b;
       b = rem;
       rem = myMod(a, b);
   }
   cout<<"GCD is "<<b<<endl;
}
int myMod(int a, int b){
   if(a<b)
       return a;
   float x;
   x = float(a)/b;
```

```
    x = x-int(x);
    return round(x*b);
}
```

Q2 (a) Definitions (1.5+1.5), Examples (1.5+1.5)
Q2 (b)

```
#include <iostream>
#include<cmath>
using namespace std;
int noOfDaysInMonth(int , int);
bool isLeapYear(int);
int main(){
    int day, month, year, day1;
    cout<<"Enter the day (1-31)"<<endl;
    cin>>day;
    cout<<"Enter the month (1-12)"<<endl;
    cin>>month;
    cout<<"Enter the year1"<<endl;
    cin>>year;
    day = day + 25;
    day1 = noOfDaysInMonth(month, year);
    if (day > day1){
        month = month + 1;
        day = day%day1;
    }
    if (month > 12){
        month = 1;
        year = year+1;
    }
    cout<<"Date after 25 days is:
"<<day<<"/"<<month<<"/"<<year<<endl;
    return 0;
}
int noOfDaysInMonth(int mon, int year){
    if (mon == 4 || mon == 6 || mon == 9 || mon == 11)
        return 31;
    if (mon ==2){
        if (isLeapYear(year))
```

```
            return 29;
        else
            return 28;
    }
    return 31;
}
bool isLeapYear(int year){
    if (((year % 4 == 0) && (year % 100 != 0)) || (year % 400
== 0))
        return true;
    else
        return false;
}
```

Q3. Marks to be given in both cases:(positives followed by zeros followed by negatives OR negatives followed by zeros followed by positives)

```
#include<iostream>
using namespace std;
void COUNT(int* arr)
{
     int p=0,n=0,z=0;
     for(int i=0;i<10;i++)
     {
      if(arr[i]>0)   p++;
      else if (arr[i]<0) n++;
      else z++;
    }
     cout<<"\n\n Total number of Positive Integers in intlist=
"<<p;
     cout<<"\n\n Total number of Negative Integers in intlist=
"<<n;
     cout<<"\n\n Total number of Zeros in intlist= "<<z;
}

void SORTLIST(int* arr)
{
    int temp[10];
    for(int i=0;i<10;i++)
    temp[i]=arr[i];

    int idx=0;

    for(int i=0;i<10;i++)
     {
      if(temp[i]>0)
       {
         arr[idx]=temp[i];
         idx++;
       }
     }



     for(int i=0;i<10;i++)
     {
      if(temp[i]==0)
       {
         arr[idx]=temp[i];
         idx++;
       }
     }



     for(int i=0;i<10;i++)
     {
      if(temp[i]<0)
       {
         arr[idx]=temp[i];
         idx++;
       }
     }

}

int main()
```

```
{
      int intlist[10];
      cout<<"Enter 10 integers \n";
      for(int i=0;i<10;i++)
      cin>>intlist[i];
      cout<<"\n The elements of intlist are \n";
      for(int i=0;i<10;i++)
      cout<<intlist[i]<<" ";

      COUNT(intlist);

      SORTLIST(intlist);

      cout<<"\n The elements of intlist after sorting are \n";
      for(int i=0;i<10;i++)
      cout<<intlist[i]<<" ";
      return 0;
}
```

Q4.

```
/*******************************************************************
          Matrix.h

*******************************************************************/

#ifndef MATRIX_H
#define MATRIX_H
#include<iostream>
using namespace std;
class Matrix
{
  private :
     double a;
     double b;
     double c;
     double d;
  public :
   Matrix();
   Matrix(double w, double x, double y, double z);
   double getA() const;
   double getB() const;
   double getC() const;
   double getD() const;
```

```
    void setA(double w);
    void setB(double x);
    void setC(double y);
    void setD(double z);
    void setW(double w, double x, double y, double z);
    Matrix operator+(const Matrix& that) const;
    Matrix operator*(const Matrix& that) const;

};

ostream& operator<<(ostream& os, const Matrix q);


#endif // MATRIX_H

/*******************************************************************
                Matrix.cpp
*******************************************************************/

#include<iostream>
#include "Matrix.h"

using namespace std;

Matrix::Matrix()
{
    a = 0.0;
    b = 0.0;
    c = 0.0;
    d = 0.0;
}

Matrix::Matrix(double w, double x, double y, double z)
{
    a = w;
    b = x;
    c = y;
    d = z;
}

double Matrix::getA() const
{
    return a;
}
double Matrix::getB() const
{
    return b;
}
```

```
double Matrix::getC() const
{
   return c;
}
double Matrix::getD() const
{
   return d;
}
void Matrix::setA(double w)
{
   a = w;
}
void Matrix::setB(double x)
{
   b = x;
}
void Matrix::setC(double y)
{
   c = y;
}
void Matrix::setD(double z)
{
   d = z;
}

void Matrix::setW(double w, double x, double y, double z)
{
   a = w;
   b = x;
   c = y;
   d = z;
}
Matrix Matrix::operator+(const Matrix& that) const
{
   Matrix temp;

   temp.a = a + that.a;
   temp.b = b + that.b;
   temp.c = c + that.c;
   temp.d = d + that.d;

   return temp;
}

Matrix Matrix::operator*(const Matrix& that) const
{
   Matrix temp;
   temp.a = (a * that.a) + (b * that.c);
   temp.b = (a * that.b) + (b * that.d);
   temp.c = (c * that.a) + (d * that.c);
   temp.d = (c * that.b) + (d * that.d);

   return temp;
}


ostream& operator<<(ostream& os, const Matrix q)
{
   os << q.getA() << "   " << q.getB() << " \n"
     << q.getC() << "   " << q.getD() << endl;

   return os;
}


/*****************************************************************
            clientMatrix.cpp


      Create a Matrix class to represent an element of the ring

      M2(R)={[a b]
           [c d]  |  a, b, c, d in R}

      with respect to the usual matrix addition and multiplication.

      The class should have the following features :

      1. The Matrix class should have four private data elements
      a, b, c and d (of double type) to represent the 2x2 matrix
      A with a = A11, b = A12, c = A21,  d = A22 :

          A =   [a b]
                [c d]

      2. The class should include the following constructors :

      (a) A default constructor to represent the zero 2x2 matrix
      (b) A four argument constructor

      3. Include get methods to inspect the values held in a, b, c
      and d.

      4. Include set methods to change the values held in a, b, c
      and d.
```

5. Include the operator + to add two Matrix objects

6. Include the operator * to multiply two Matrix objects

7. Include an operator << for printing a Matrix object in two row form :

```
[a b]
[c d]
```

Also, write a driver/client program to test the class Matrix.

```
*******************************************************************/

#include<iostream>
#include "Matrix.h"

using namespace std;

int main()
{
   Matrix X, Y(1, 2, 3, 4);

   cout << "X = \n" << X << endl;
   cout << "Y = \n" << Y << endl;

   cout << "X + Y = \n" << X + Y << endl;

   X.setW(2, 4, 5, -8);

   cout << "X = \n" << X << endl;

   X.setW(1, 2, 3, -4);
   Y.setW(-2, 1, 2, 5);

   cout << "X = \n" << X << endl;
   cout << "Y = \n" << Y << endl;

   cout << "X * Y = \n" << X * Y << endl;
   cout << "Y * X = \n" << Y * X << endl;

   return 0;
}
```

Q5.

```
 #include<iostream>
```

```
using namespace std;

pair<int, int>    COUNTCASE(string s)
{
        pair<int, int> c;
        c.first=0;         //to count lowercase alphabets
        c.second=0;   //to count uppercase alphabets
        for(int i=0;i<s.size();i++)
        {
                if(int(s[i])>=97 && int(s[i])<=122)       c.first++;
                if(int(s[i])>=65 && int(s[i])<=90)        c.second++;
        }

        return c;
}

pair<bool, bool>   CHECKSPECIAL(string s)
{
        pair<bool, bool> c;
        c.first=false;          //to check ! character
        c.second=false;   //to check ? character
        for(int i=0;   i< s.size();   i++)
        {
                if(s[i]=='!')    c.first=true;
                if(s[i]=='?')   c.second=true;
        }

        return c;
}


int main()
{
        string str;
        cout<<"Enter a sentence here\n";
        getline(cin,str);
        cout<<"\nYou entered: "<<str;

        pair<int, int> count;
        count=COUNTCASE(str);
        cout<<"\nNumber of Lowercase alphabets= "<<count.first;
        cout<<"\nNumber of Uppercase alphabets= "<<count.second;

        pair<bool, bool>check;
        check=CHECKSPECIAL(str);
        if(check.first==true)                cout<<"\nSpecial Character '!' FOUND in the
sentence";
```

```
        else                                cout<<"\nSpecial Character '!' NOT FOUND in
the sentence";
        if(check.second==true)              cout<<"\nSpecial Character '?' FOUND in the
sentence";
        else                                cout<<"\nSpecial Character '?' NOT FOUND in
the sentence";

        return 0;
}
```

Q6.

a.

```
/*******************************************************************************************
*****
            factorial.cpp

            Write a program in C++ to find 100! using a procedure with the help
            of GMP package.
********************************************************************************************
***/


#include<iostream>
#include "gmpxx.h"
#include "gmp.h"
using namespace std;
mpz_class factorial(mpz_class x)
{
   mpz_class I;
   mpz_class temp = 1;

   for(I = 1; I <= x; I++)
   {
      temp = temp * I;
   }

   return temp;
}


int main()
{
   mpz_class n,  fact;
   cout << "enter an integer of any size " << endl;
   cin >> n;
   fact = factorial(n);
   cout << "the factorial of " << n << " is " << fact << endl;
   return 0;
}
```

b.

```
/*********************************************************************************
   Write a C++ program to generate 50 numbers randomly from the set
   {1, 2, ... , 10} and write them to file named random.txt.

   Then the program must find the average of the 50 numbers from the
   file random.txt  and write the average to a file named average.txt.
*******************************************************************************/

#include<iostream>
#include<fstream>
#include<cstdlib>
#include<ctime>

using namespace std;

int main()
{
   int k;
   ofstream out;

   out.open("random.txt");

   srand(time(0));
   int temp;

   for(k = 1; k <= 50; k++)
   {
      temp = rand() % 10 + 1;
      out << temp << "   ";
   }

   out.close();

   ifstream inp;
   inp.open("random.txt");
   ofstream outA;
   outA.open("average.txt");

   double sum = 0.0;

   for(k = 1; k <= 50; k++)
   {
      inp >> temp;
      sum = sum + temp;
   }
```

```
    outA << "average is " << (sum / 50) << endl;

    inp.close();
    outA.close();


    return 0;
}
```