Problem (1) [10 points] Write (in C++) a “Fraction” class with two integers to store the numerator and denominator. Also write a non-default constructor to initialize the fraction. Write the class following the proper coding conventions.

class Fraction
{
private:
    int numerator;
    int denominator;
public:
    Fraction(int n, int d);
};

Fraction::Fraction(int n, int d)
{
    numerator = n;
    denominator = d;
}
Problem (2) [10 points] Suppose there is a “Course” class. For each of the following pieces of code, describe whether the operator overloading can be done with: (1) a friend function only, (2) a member function only (a function inside the “Course” class), or (3) both types of functions would work (with member functions part of the “Course” class).

Provide a sentence justifying your reasoning.

Course c;

1:
cout << c;

2:
if("csci1113" == c)

3:
UMNrequirement libEd;
libEd += c;

1: Friend function only.
   cout is on the left, so if it was a member function we would need to change the istream class (which we cannot)
2: Friend function only.
   "csci1113" is a char array not a class.
   As this is on the left it must be a friend function.
   (You can treat "csci1113" as a string, but the answer is still the same).
3: Friend function only.
   The problem description says we are interested with the member function being inside the "Course" class.
   As the Course object ("c") is on the right, this cannot be a member function in "Course".
   (Partial credit given if you said it can be a member function inside "UMNrequirement class")
Problem (3) [10 points] What is the output of the following code:

class A{
    protected:
        int x;
    public:
        A();
        A(int in);
        int getValue();
        virtual int virtualValue();
    A::A() {
        x = 1;
    }
    A::A(int in) {
        x = in;
    }
    int A::getValue() {
        return x;
    }
    int A::virtualValue() {
        return x;
    }

class B : public A{
    public:
        B(int in);
        int getValue();
        virtual int virtualValue();
    B::B(int in) {
        x = in;
    }
    int B::getValue() {
        return 2*x;
    }
    int B::virtualValue() {
        return 2*x;
    }

int main() {
  A a = A(5);  B b = B(5);
  A* aptr = &a;
  A* bptr = &b;
  cout << a.getValue() << " " << b.getValue()  << endl;
  cout << aptr->getValue() << " " << bptr->getValue()  << endl;
  cout << a.virtualValue() << " " << b.virtualValue()  << endl;
  cout << aptr->virtualValue() << " " << bptr->virtualValue()  << endl;
}
Problem (4) [10 points] What are both the variables and their values inside “thing” from main()? How many constructors are run to create “thing”?

class A{
protected:
   int a;
   int b;
   int c;
public:
   A();
   A(int x);
};

A::A() {
   a = 1;
   b = 2;
   c = 3;
}

A::A(int x) {
   a = x;
   b = x+1;
   c = x*2;
}

class B : public A{
private :
   int d;
public :
   B(int x);
};

B::B(int x) {
   d = x;
}

int main() {
   B thing = B(5);
}

What are both the variables and their values inside “thing” from main()? How many constructors are run to create “thing”? 
Variables:
  a = 1
  b = 2
  c = 3
  d = 5

Constructors:
  2 (non-default B constructor, default A constructor)
Problem (5) [10 points] Suppose you have a class “Linker” as shown below. Write (in C++) code below the lines in main to change this from a chain into a ring by adding the dashed arrow link in the following picture:

class Linker {
    public Linker* p;
};

int main() {
    Linker start;
    // setup pointers, add code below
}

    Linker* current = &start;
    while(current->p != NULL)
    {
        current = current -> p;
    }
    current -> p = &start;


Problem (6) [10 points] Write (in C++) a “<” operator for the “Date” class below to indicate whether a date comes before another date. Write both the function declaration (inside the class) along with the operator’s function definition (how it works).

class Date {
public:
    int month;
    int day;
    int year;
};

int main()
{
    Date cppDayFinal;
    Date cppNightFinal;
    // initialize ^^ (both)
    if(cppNightFinal < cppDayFinal)
    {
        cout << "FIRST" << endl;
    }
}

class Date {
public:
    int month;
    int day;
    int year;
    bool operator<(Date right);
};

bool Date::operator<(Date right)
{
    if(year < right.year)
    {
        return true;
    }
    else if(year == right.year)
    {
        if(month < right.month)
        {
            return true;
        }
else if(month == right.month)
{
    if(day < right.day)
    {
        return true;
    }
    else
    {
        return false;
    }
}
else
{
    return false;
}
}
else
{
    return false;
}
}

... OR ...

class Date {
public:
    int month;
    int day;
    int year;
    friend bool operator<(Date left, Date right);
};

bool operator<(Date left, Date right)
{
    int leftVal = left.year*1000000+left.month*1000+left.day;
    int rightVal = right.year*1000000+right.month*1000+right.day;
    return leftVal < rightVal;
}
Problem (7) [10 points] Make a copyArray() function that takes as input two things:
(1) a dynamically created array of “Complex” numbers (assume there exists a “Complex”
class), and
(2) the size of this array.
This function should return a copy of the dynamically created array (i.e. changing the
returned array should not effect the array passed in).

Complex* copycat(Complex inArray[], int size)
{
    Complex* result = new Complex[size];
    for(int i=0; i < size; i++)
    {
        result[i] = inArray[i];
    }
    return result;
}
Problem (8) [10 points] Find 3 errors in the code below. Assume that the code is completely shown except for #includes and “using namespace std”. For each error, identify whether it is a runtime error, syntax error or logic error. You must also precisely describe why you think the part of code you identify is an error.

class Temperature {
    double* days;
    int size;
    Temperature();
};

temperature::Temperature()
{
    days = new int[31];
    size = 31;
}

int main() {
    Temperature dec();
    // do something with dec
}

1. syntax or logic:
   no access modifiers (i.e. "public:" or "private:" ) in class Temperature
2. syntax:
   days is a double* not int* so constructor cannot create int array
3. logic:
   main is declaring a function dec() not a variable dec using the default constructor
4. logic:
   no deconstructor so memory leak
Problem (9) [10 points] What is the output of this code:

```cpp
for(int i=0; i < 5; i++) {
    for(int j=0; j < 8; j++) {
        if(j%3 == 0) {
            cout << "X";
        } 
        else if(i == 2 && j < 4) {
            cout << "X";
        } 
        else if( (i == 0 || i == 4) && j > 4) {
            cout << "X";
        } 
        else {
            cout << " ";
        }
    }
    cout << endl;
}
```

X  X  XXX
X  X  X
XXXX  X
X  X  X
X  X  XXX
**Problem (10)** [10 points] Write a *recursive* function that takes as input 3 things: an integer array, the array’s length and a number. Have this recursive function display all the indexes in the array that have the value of this number. You may pass more arguments into the recursive function in addition to the three above if you wish.

Example usage:
```c
int x[] = {1, 2, 3, 1, 2, 1, 1};
recursivePrint(x, 7, 1);
// above couts: 0, 3, 5, 6
```

```c
def recursivePrint(int arr[], int len, int toFind)
{
    if(len==0)
    {
        return;
    }
    else
    {
        recursivePrint(arr, len-1, toFind);
        if(arr[len-1] == toFind)
        {
            cout << (len-1) << " ", ";
        }
    }
}
```
Problem (11) [10 points] Write (in C++) a function to compute the circumference of a trapezoid. The formula for this is: \( b_1 \cdot b_2 \cdot h \cdot \left( \frac{1}{\sin(A)} + \frac{1}{\sin(B)} \right) \). Note: the “cmath” library has a sin() function.

```cpp
double itsatrap(double b1, double b2, double h, double A, double B)
{
    return b1*b2*h*(1/sin(A) + 1/sin(B));
}
```
**Problem (12)** [10 points] Read a sentence from the keyboard, then re-display all parts of that sentence that are in quotes. You can assume there will be an even number of quotes.

Example input:
My "friend" ate "something" that tasted "unknown".

Example output:
friend something unknown

```cpp
string sent;
getline(cin, sent);
bool toPrint = false;
for(int i=0; i < sent.length(); i++)
{
    if(sent[i] == '"' && toPrint)
    {
        cout << " ";
        toPrint = false;
    }
    else if(sent[i] == '"' && !toPrint)
    {
        toPrint = true;
    }
    else if(toPrint)
    {
        cout << sent[i];
    }
}
```