The SafeArray Project

Recall that there is no check on an array index out of bounds. However, during program execution, an array index out of bounds can cause serious problems. Also, in C++, the array indexes starts at 0.

Design and implement the templated `class SafeArray` that solves the array index out of bounds problem and also allows the user to begin the array indexes to start at an arbitrary index a and end at an arbitrary index b-1, provided a < b. These integers may be negative

Every object of this class represents an array of values of the template type. The templated class should be contained in a file named *SafeArray.h*. That MUST be the name with no modifications.

The private section of the class declaration is as follows:

```
  int firstValid;
  int firstInvalid;
  T *A;
```

You will use a dynamic array A to implement the SafeArray objects. Moreover, the user will be able to use indexes i with

```
        firstValidIndex <= i < firstInvalidIndex.
```

This means that your code for the [ ] operator will have to translate these indices to ones that start at 0.

Consider the following statements:

```
SafeArray<int> myArray(5);          //Line 1 firstValidIndex 0
                                    //       firstInvalidIndex 5

SafeArray<int> myMyArray(2, 13);    //Line 2 firstValidIndex 2
                                    //       firstInvalidIndex 13

SafeArray<int> yourArray(-5, 9);    //Line 3 firstValidIndex -5
                                    //       firstInvalidIndex 9
```

The statement in Line 1 declares `list` to be an array of 5 components, the component type is `int`, and the valid index references are:

```
    myArray[0], myArray[1], ..., myArray[4].
```

The statement in Line 2 declares `myArray` to be an array of 11 components, the component type is `int`, and the valid index references are:

```
    myMyArray[2], myMyArray[3], ..., myMyArray[12].
```

The statement in Line 3 declares `yourArray` to be an array of 14 components, the component type is `int`, and the valid index references are:

```
   yourArray[-5], yourArray[-4], ..., yourArray[0], ..., yourArray[8].
```

During execution, when accessing an array component, if the index is out of bounds, the program must terminate with an appropriate error message.

A reference to `yourArray[10]` should result in the following message

```
   SafeArray index 10 out of bounds: too small
```

A reference to `yourArray[-6]` should result in the following message

```
   SafeArray index -6 out of bounds: too small
```

In both of the above cases, the program should be terminated. Note: to use the `exit` function, you must include the header `<cstdlib>`.

You should also write a program to test the `SafeArray` class. Your program should have the name *SafeArrayT*est.cpp. You will submit the files *SafeArray.h* and *SafeArrayTest.cpp*. Do not submit a zip or rar file; submit two files.

An incomplete declaration of the class is shown on the next page. Since you are designing a templated class, the method implementations must be included in the file *SafeArray.h.* You are to supply the missing declarations and code in that file.

```
//------------------ file SafeArray.h ---------------------
// Declaration

template class<T>
class SafeArray
{
 public:
  SafeArray<T> myArray(int firstInvalidIndex);
  SafeArray<T> myArray(int firstValidIndex, int firstInvalidIndex);
  T& operator[](int ndx);
  void print();
  int length();
  // fill in the declarations as indicated below

  // the copy constructor below:

  // the overloaded assignment operator:

  // the destructor:

 private:
  int firstValid;
  int firstInvalid;
  T *A;
};

// Write below the implementations of the three constructors, the
// destructor, the overloaded assignment and [ ] operators, the print
// method and the length method
```