İTÜ 09.05.2005
Computer Engineering Department

Object Oriented Programming 3rd Homework

Question 1:

Do not use containers and algorithms of the STL, while solving this problem.

a) Write a template function (algorithm) "**smallest**", which takes the start address and the length (number of elements) of an array and returns the indices of the smallest element. Note that this function can operate on arrays of any type. Example:

```
int inta[4]={1,-7,0,3};                // An integer array with 4 elements
unsigned int smallest_i = smallest(inta,4);
cout << "Smallest element is :" << inta[smallest_i] << endl;
```

b) Design and write a template class **TempArray**, which is an array that can contain elements of any type. Objects of TempArray may have different sizes. The size of the array is a parameter of the constructor. The class TempArray must include necessary members so that the algorithm smallest can operate on it as shown below:

```
 TempArray<int> intArr(5);  // An array of integers with 5 empty palces
 try{
  intArr[0]= 1;
  intArr[1]= -5;
  intArr[2]= 8;
  }
 catch(const string & msg){  // indices may be out of bounds
    cout << msg << endl;
 }
 unsigned int smallest_i = smallest(intArr,3);
 cout << "Smallest element is :" << intArr[smallest_i] << endl;
```

c) Design and write a class **Binary** that can be used to define unsigned binary integer numbers. The constructor of the class will take the number of bits (the length of the binary number) as a parameter. In the constructor, the value of each bit will be read from the keyboard.
The class Binary must include necessary members so that it can be used with the template class TempArray and algorithm smallest as shown below. The class Binary also includes a method calculate that calculates and returns the value of the binary number as a decimal integer number.

```
TempArray<Binary> binArr(3);          // an array for three binary numbers
 try{
  Binary b1(3), b2(2);              // Two binary numbers with 3 and 2 bits
  binArr[0] = b1;
  binArr[1] = b2;
 }
 catch(const string & msg){            // indices may be out of bounds
    cout << msg << endl;
 }
 unsigned int smallest_i = smallest(binArr,2);
 cout << binArr[smallest_i] << endl;   // Print the smallest binary number
```

SUBMISSION:

- Read **C++ Programming Standards**, which can be found on the course web page. Write your program according to these rules. Don't forget to include sufficient explanation into your code.
- The name of the source file should be your number. For example: 040000601.cpp
- Copy the source file via anonymous ftp to akasya.cs.itu.edu.tr in to directory oop/ between **9.00** and **17.30** on **23rd May 2005**.
- List files and check the size of your file.
- You cannot view, delete, modify, overwrite or get files from this directory (Even your homework).
- Late submitted homeworks are not accepted.
- **Cheating** will not be tolerated. If cheating is discovered, all responsible students will be punished. Punishment for cheating is a score of **–50/100** points.
  It is allowed to discuss how to solve a problem with your classmates; however, **homeworks are not group homeworks. All actual programming should be an independent effort**.