

**Due: Feb 10, Wednesday**

Consider the following definition of class `myArray`: (defined in `myArray.h`)

```
class myArray
{
public:
    myArray();                // Default constructor;
    myArray(int s, int iv);    // s:size; iv:initial values;
    myArray(const myArray & a);    // Copy constructor;
    myArray & operator = (const myArray & a);    // Overload Assignment = ;
    ~myArray();                // Destructor;
    int & operator [] (int i);    // Return the ith element in the array;
    double average();            // Return the average of the numbers in the array;
    int max();                    // Return the biggest number's index in the array;
    int min();                    // Return the smallest number's index in the array;
    int getSize();                // Return the size of the array;

private:
    int size;
    int *A;
};
```

The `[]` operator is implemented as follows:

```
int & myArray::operator[] (int i)
{
    if ( i < 0 || i >= size)
        { cout << "Out of range:"; return A[0]; }
    return A[i];
}
```

1. Make directory `~/IT279/Asg1/` on your Unix account for this assignment, where `~` denotes your home directory.
2. Download `myArray.h` and `Asg1.cpp` from my unix public directory `/home/cli2/public/IT279/`. Implement class `myArray` in c++ program `myArray.cpp`.
3. Define the following two function in your `myArray.cpp`, but they are not members nor class functions of class `myArray`.

- `myArray fibonacci(int n);`

For any integer  $n \geq 0$ , `fibonacci(n)` returns a `myArray` with  $n$  fibonacci numbers,  $f_0, f_1, f_2, \dots, f_{n-1}$  stored in it in the same order, i.e. the internal array `A[i]` stores  $f_i$ . The fibonacci numbers are defined as follows:

$$f_0 = 0, f_1 = 1, \text{ and for } n \geq 2, f_n = f_{n-1} + f_{n-2}.$$

For example, if we have

```
myArray fib = fibonacci(20);
```

Then, `fib[0]` =  $f_0$ , `fib[1]` =  $f_1$ , ..., and `fib[19]` =  $f_{19}$ .

- `void rand(myArray *ma);`

This function will take a pointer of `myArray` and replace every entry with a random integer between 0 and the size of the `myArray` pointed by `ma` in the argument. Pay attention to the pointer type used for the parameter, and make sure your program actually changes the contain of `*ma`.

4. The standard output of this assignment is the output of `Asg1.cpp`. Note that you have to write your own test program to make sure every function is correct. In addition to `Asg1.cpp`, I will run my own test program to see if there are hidden errors (including memory leaking). Any mistakes will cost you some points.

### Secret directory under public

After you've finished your work, or have decided that what you have is the final version for me to grade, you have to copy all your programs to a secret directory under your `public/IT279myWork` so I can grade them (i.e., compile, run, and check the codes). Select a secret name, say "peekapoo" as an example (you should chose your own), and that will be the secret directory.

You can simply copy my `/home/cli2/public/IT279/copy279all.sh` and use the following command line to run it. (You can put `copyall.sh` in any directory.)

```
bash copy279all.sh peekapoo
```

### Final Words:

You have to follow the submission guidelines, i.e., cover page (that contains assignment number, student's names, student **ULID**, and secret directory), summary, source code(optional), output, folder, and so on. **No late work will be accepted.**