

Programming Assignment 9

Prof. Zhang

Fall 2018

This lab practices implement a class that uses **dynamic array** to store a collection of values of same type, i.e., a list.

Note that in this lab, you are asked to implement the partially filled array using an accompanying variable `len` to keep track how many slots of the array is filled (see below).

```
class IntList{
public:
    ....
private:
    int* array; //address of dynamically allocated array
    int size;  //size of the array
    int len;   // number of int elements stored in array
              //array is partially filled from lower index on, i.e.,
              //array[0], array[1], ... array[len-1] are occupied,
              //while array[len], ... array[size-1] are vacant
};
```

Requirement

Extend the given code to support a `IntList` class. Run the following command after logging onto storm to copy the provided code (`IntList.h`, `IntList.cpp`, `main.cpp`):

CopyIntList

An object of `IntList` class stores a list int values in an array, and provide member functions for

1. One constructor with default argument value of 10: allocate the array of the given size in the heap

```
IntList(int size=10)
```

2. Copy constructor that takes an IntList object as parameter and initializes an IntList object to be a copy of src

```
/* Initialize an IntList object to be a copy of the src object passed in the parameter
   Initialize the object's size and len member variable to be same as those of src
   object,
   Allocate memory for an array of same size and stores its address in array
   Copy all elements stored in src.array to current object (that is being initialized)
   */
IntList (const IntList & src)
```

3. destructor: deallocates the array that is allocated in the constructor

```
/* deallocate the dynamic array (whose address is stored at array member
   variable) */
~IntList()
```

4. push_back(int v): Add a new int value v into the list
5. search (int v): Search for a value in the list, return true if it's found; return false if it's not.
6. sort(): Sort the int values in the list (array) into ascending order (you can use bubble sort or selection sort algorithms)
7. Overload the following operators:
 - A. output operator <<: This should output the size of the array used in the IntList, and also the elements in the array.
 - B. index operator []: ~~this way, listA[i] returns the i-th int variable in the listA; if there is no i-th element, exit the program with error~~ **(given)**
 - C. (extra credit) override + operator to merge two IntList objects
 - D. (extra credit) override - operator to remove all ints in second list from the first list

- E. (extra credit) override = operator as **member function** to assign one IntList object to another

```
// if calling object is the same as src, do nothing and return src
// ow., deallocate calling object's array
// the rest is similar to copy constructor: copy len, size member variable
// allocate a new array and copy src.array elements to calling object's array
IntList& operator= (const IntList & src);
```

Testing and main

Your main should have the following testing code minimally:

```
//.. #include etc omitted
int main()
{
    IntList listA(); // calling zero-parameter constructor
    IntList listB(30); // calling constructor with one int parameter

    // call push_back to put 0, 1, ...,9 into listA
    for (int i=0;i<10;i++)
        listA.push_back (i);
    cout <<listA <<endl; // output operator

    cout <<"Enter a value, and I will look it up in listA:";
    int value;
    cin >> value;
    if (listA.search(value))
        cout <<value<<" is in the listA\n";
    else
        cout <<value <<" is not in the listA\n";

    //Similarly, test delete(), sort()

    int index;
    cout <<"Enter the index of list to change:";
    cin >> index;
    cout <<"What new value:";
```

```
cin >> value;
listA[index] = value; // this will modify the index-th element in the listA

listB = listA; // assignment operator testing
cout <<listB<<endl;

//destructor will be called when main() ends, on all IntList objects
}
```

To submit:

```
submit2000 LAB9 IntList.h
submit2000 LAB9 IntList.cpp
submit2000 LAB9 main.cpp
submit2000 LAB9 Makefile
```