

Problem D: Array Template

(25% related to Lab 12)

Problem Description

Modify the Array class in Fig. 11.6~Fig. 7 into a class template that can be used to create an array of any data type. The array should give an exact number of entries to store data. That is, there is no empty entry. At the same time, add the following member functions to the template.

Array<T>& insertTop(T &): insert the element specified by the parameter into the array as the last element.

Array<T>& deleteTop(): delete the last element in the array.

Array<T>& operator+(Array<T> &B): concatenate two arrays by placing B behind the underlying array. For example if A={1, 2, 3, 4} and B={5, 6, 7}, then A+B will return {1, 2, 3, 4, 5, 6, 7} and B+A will return {5, 6, 7, 1, 2, 3, 4}.

Array<T>& operator!(): reverse the order of the elements in the underlying array. For example, if A={1, 2, 3, 4}, then !A will return A={4, 3, 2, 1}

Input format

The input to the program will be the data stored in an array. Input the number of items based on the prompting message.

Output format

The output should be exactly same as the printout in the example output.

Requirements

The main() function in Fig. 11.8 has been modified and given to you below. You should not modify it further.

```
int main()
{
    Array<int> integers1( 7); // seven-element Array
    Array<int> integers2; // 10-element Array by default

    cout << "Enter 17 integers:" << endl;
    cin >> integers1 >> integers2;

    cout << "\nAfter input, the Arrays contain:\n"
        << "integers1: \n" << integers1
        << "integers2: \n" << integers2;

    // use overloaded inequality (!=) operator
    cout << "\nEvaluating: integers1 != integers2" << endl;

    if ( integers1 != integers2 )
        cout << "integers1 and integers2 are not equal" << endl;
    cout << "\n\nAssigning 1001 to integers2[8]" << endl;
    integers2[ 8 ] = 1001;
    cout << "integers2:\n" << integers2 << endl;
    integers2.insertTop(222);
    integers2.insertTop(222);
    cout << "After insertion and deletion: integers2:\n" << integers2<<endl;
```

```
integers2.deleteTop();
cout << "After insertion and deletion: integers2:\n" << integers2<<endl;
Array<int> integers3 = integers1 + integers2;
cout << "Integers3 :\n" << integers3 <<endl;
cout << "integers1:\n" << !integers1<< endl;

Array<double> double1(12); // invokes copy constructor
cout << "\nEnter 12 double precision numbers:" << endl;
cin >> double1;
cout << "\nCreate double2 initialized with double1: " << endl;
Array<double> double2(double1); // note target Array is smaller
cout << "double1: \n" << double1 << "double2: \n" << double2;
// use overloaded equality (==) operator
cout << "\nEvaluating: double1 == double2" << endl;
if ( double1 == double2 )
    cout << "double1 and double2 are equal" << endl;
// use overloaded subscript operator to create lvalue
cout << "\n\nAssigning 100.01 to double1[6]" << endl;
double1[ 6 ] = 100.01;
cout << "double1:\n" << double1 << endl;
double1.insertTop(3333.3);
double1.deleteTop();
double1.deleteTop();
cout << "After insertion and deletion: double1:\n" << double1<<endl;
Array<double> double3 = double1 + double2;
cout << "double3 :\n" << double3<<endl;
cout << "double1:\n" << !double1 << endl;

Array<string> strA(5);
cout << "\nEnter 5 strings:" << endl;
cin >> strA;
strA.insertTop("Programming");
strA.insertTop("C++");
cout << "After insertion and deletion: strA:\n" << strA<<endl;
// attempt to use out-of-range subscript
Array<string> strB = strA + strA;
cout << "strB :\n" << strB<<endl;
cout << "!strA:\n" << !strA << endl;
strA.deleteTop();
cout << "!strA:\n" << !strA << endl;

return 0;
} // end main
```

Example Input:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2
aa bb cc dd ee

Example Output (containing input):

input

```
Enter 17 integers:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

After input, the Arrays contain:
integers1:
    1         2         3         4
    5         6         7
integers2:
    8         9         10        11
   12        13        14        15
   16        17

Evaluating: integers1 != integers2
integers1 and integers2 are not equal

Assigning 1001 to integers2[8]
integers2:
    8         9         10        11
   12        13        14        15
  1001        17

After insertion and deletion: integers2:
    8         9         10        11
   12        13        14        15
  1001        17        222       2222

After insertion and deletion: integers2:
    8         9         10        11
   12        13        14        15
  1001        17        222

Integers3 :
    1         2         3         4
    5         6         7         8
    9        10        11        12
   13        14        15       1001
   17       222

integers1:
    7         6         5         4
    3         2         1
```

input

```
Enter 12 double precision numbers:
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2

Create double2 initialized with double1:
double1:
    0.1      0.2      0.3      0.4
    0.5      0.6      0.7      0.8
    0.9      1      1.1      1.2
double2:
    0.1      0.2      0.3      0.4
    0.5      0.6      0.7      0.8
    0.9      1      1.1      1.2

Evaluating: double1 == double2
double1 and double2 are equal

Assigning 100.01 to double1[6]
double1:
    0.1      0.2      0.3      0.4
    0.5      0.6      100.01    0.8
    0.9      1      1.1      1.2

After insertion and deletion: double1:
    0.1      0.2      0.3      0.4
    0.5      0.6      100.01    0.8
    0.9      1      1.1

double3 :
    0.1      0.2      0.3      0.4
    0.5      0.6      100.01    0.8
    0.9      1      1.1      0.1
    0.2      0.3      0.4      0.5
    0.6      0.7      0.8      0.9
    1      1.1      1.2

double1:
    1.1      1      0.9      0.8
    100.01    0.6      0.5      0.4
    0.3      0.2      0.1

Enter 5 strings:
aa bb cc dd ee

After insertion and deletion: strA:
    aa      bb      cc      dd
    ee Programming      C++

strB :
    aa      bb      cc      dd
    ee Programming      C++      aa
    bb      cc      dd      ee
    Programming      C++

!strA:
    C++ Programming      ee      dd
    cc      bb      aa

!strA:
    bb      cc      dd      ee
    Programming      C++
```

input