Problem D: Double-Subscripted Arrays with opeartor()

(30% for part I, 15% for Part II, related to Lab 12 & Lab 13)

Problem Description

This problem has two parts. The first part is to modify the **Array** *class* given in Fig. 11.6 and Fig. 11.7 into an **Array** *template*. At the same time, extend it to create a two dimensional array with two subscripts and overload the operator ( ) to accept two subscripts (called *double-subscripting*). Hence, instead of saying

chessBoard[row][column]

for an array of objects, overload the function call operator to allow the alternate form

chessBoard(row, column)

The **Array** template should allow us to create an array of any number of rows and any number of columns. The subscripts of row and column both start from zero. The template should supply operator () to perform double-subscripting operations. For example, in a 3-by-5 double-subscripted array, called *ary*, a programmer could write *ary*(1, 3) to access the element at row 1 and column 3. Although an array is doubly subscripted, it is stored linearly as one dimensional array of $row \times column$ elements in memory. The elements should be stored in *row major* format. Function operator( ) should perform the proper pointer arithmetic to access each element of the array. There should be two versions of operators–one that returns **T &** (so that an element of a double-subscripted array can be used as an *lvalue*) and one that returns **const T &**. The template should also provide the following operators: ==. !=, =, <<(for outputting the array in row and column format) and >>(for inputting the entire array contents). The operator<< should print the array row-by-row and a row per line on the monitor.

The second part is to address the problems of memory allocation exception during creating a double-subscripted array and an out-of-range exception during accessing elements in a double-subscripted array. A memory allocation exception occurs if the program can not obtain the requested memory for creating an array. You should design your code to make the exceptions occur when they are supposed to do so.

The main() function will be provided and should not be changed. The input to the program will be first 25 integers which can appear on an arbitrary number of lines. It is then followed by 10 floating point numbers on a line, and then followed by 10 strings on a line.
The content of output should be exactly line-by-line the same as that given in the example output.

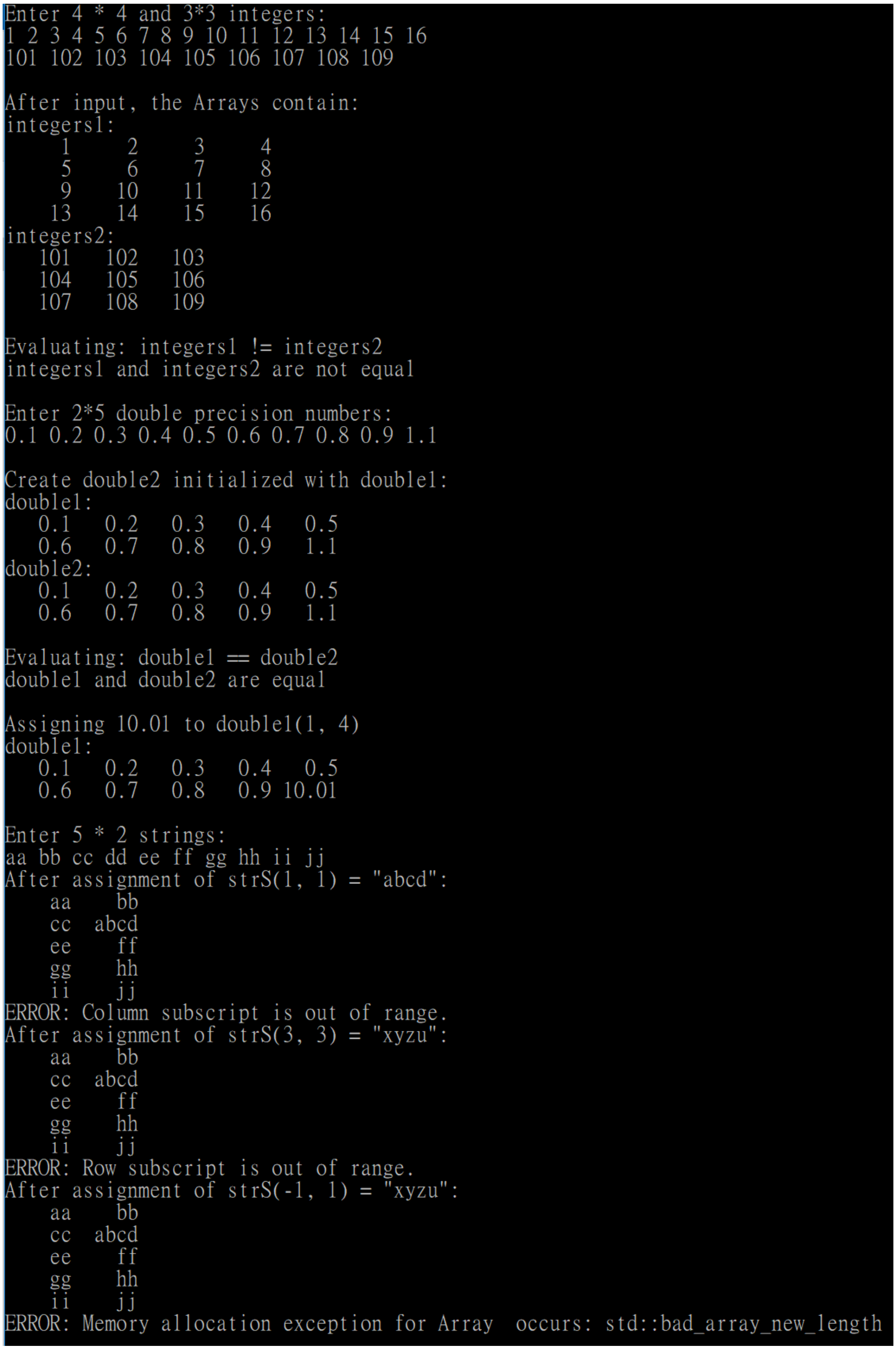
Hint for solving the problem:

- You have to know how to overload the operator<< and operator>> as a friend to Array template. You can check the lecture notes to understand how this can be done correctly.
- The code in Fig. 11.6 and Fig. 11.7 is provided.

Input:

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
101 102 103 104 105 106 107 108 109
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.1
aa bb cc dd ee ff gg hh ii jj
```

Example Output:



```
Enter 4 * 4 and 3*3 integers:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
101 102 103 104 105 106 107 108 109

After input, the Arrays contain:
integers1:
     1     2     3     4
     5     6     7     8
     9    10    11    12
    13    14    15    16
integers2:
   101   102   103
   104   105   106
   107   108   109

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

Enter 2*5 double precision numbers:
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.1

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
double2:
   0.1   0.2   0.3   0.4   0.5
   0.6   0.7   0.8   0.9   1.1

Evaluating: double1 == double2
double1 and double2 are equal

Assigning 10.01 to double1(1, 4)
double1:
   0.1   0.2   0.3   0.4   0.5
   0.6   0.7   0.8   0.9 10.01

Enter 5 * 2 strings:
aa bb cc dd ee ff gg hh ii jj
After assignment of strS(1, 1) = "abcd":
    aa    bb
    cc  abcd
    ee    ff
    gg    hh
    ii    jj
ERROR: Column subscript is out of range.
After assignment of strS(3, 3) = "xyzu":
    aa    bb
    cc  abcd
    ee    ff
    gg    hh
    ii    jj
ERROR: Row subscript is out of range.
After assignment of strS(-1, 1) = "xyzu":
    aa    bb
    cc  abcd
    ee    ff
    gg    hh
    ii    jj
ERROR: Memory allocation exception for Array  occurs: std::bad_array_new_length
```