CSCI 335
First programming assignment (100 points)
Due September 9
Follow the blackboard instructions on writing and submitting programming assignments
We will not debug your assignment. It should compile to receive any credit. It should run correctly to receive full credit.

Programming: Matrix manipulation

Create and test a class called **matrix**. For instance a two by two matrix of integers could be
[10 20
  1   0].
A three by one matrix of doubles could be
[10.0
  2.0
  50.0].
A matrix can have any dimension (i.e. N rows and M columns where N,M are integers greater than or equal to one). An empty matrix has 0 rows and 0 columns.

The purpose of this assignment is to have you create a matrix class from scratch without using the STL. In the sample code provided you can see an initial implementation of the matrix class that uses a vector. You can start with this implementation but you can’t use a vector class as a private data member. You should use pointers instead. Unless you get permission from me, don't include any libraries except iostream and cstdlib.

Pay special attention to Weiss's "big three," the destructor, copy constructor and assignment operator. Provide a template implementation. This assignment will help you revisit constructors, destructors, overloading of operators, and templates.

When your class is complete, function `TestingMatrix()` should work, with results as commented. You should **include that function as is** in your main testing file and you **should call** that function from your `main()` **.** You can include additional testing functions as well but the one shown here **has to be** included. Also note that some matrix operations are not allowed (for instance you can’t add a 2 by 2 matrix to a 3 by 3 matrix). Make sure that you catch these errors using exceptions.

Include **the following code** as is in your main test file.

```
TestingMatrix(){

matrix<int> a, b, c;              //Three empty matrices are created

cout << a.numrows() << " " << a.numcols() << endl; // yields 0 0

cin >> a;                    // User types [3 3 1 2 3 6 5 4 9 8 10]
                             // This will create a 3 by 3 matrix
                             // The first input is the number of rows, and the
                             //   second is the number of columns.
                             // The rest are the values inserted row by row
cout << a;                   // Output should be
                             // [1 2 3
                             //  6 5 4
                             //  9 8 10]
cin >> b;                    // User types [3 2 9 1 2 3 4 5]

cout << b;                   // Output should be
                             // [9 1
                             //  2 3
                             //  4 5]
c=a*b;                       // c is the product of a and b

cout << c << endl;           // The output should be:
                             // [25 22
                             //  80 41
                             //  137 93]

cout << b+c << endl;         // Output is the sum of b and c:
                             // [34 23
                             //  82 44
                             //  141 88]

matrix<int> d(5*b);          // d is initialized to 5*b

cout << d << endl;           //The output should be
                             //[45 5
                             // 10 15
                             // 20 25]

d += c;

cout << d << endl;           //The output should be
                             //[70 27
                             // 90 56
                             // 157 108]

cout << a[0][0] << endl;     //Should printout 1
cout << a[1][2] << endl;     //Should printout 4
d = a + b;                           //This should cause an exception that you
//are able to handle; The sizes of a and b don't agree.

}//End of TestingMatrix() function
```