

Xu Xikai
400440917
Xu503

1. bug detection

```
##### Invalided rows input
#failure testGetElementInvalid Test.cpp:92 testGetElementInvalid: expected == actual expected: Invalid indexes.
but was: Empty for Now
#starting testSetElementValid
#success testSetElementValid OK
#starting testSetElementInvalid
#success testSetElementInvalid OK
#starting testCopy
Program received signal SIGSEGV, Segmentation fault.
0x0040634c in Matrix::copy (this=0x68f6dc) at Matrix.cpp:191
191 copy.setElement(matrixData[j][i], j, i);
(gdb)
```

2. bug analysis:

```
Matrix copy = Matrix();
```

a)

This is using the C++ shadow copy, which only interact with references.
We need deep copy interaction.

3. Bug fix:

```
Matrix copy = Matrix(rowsNum, colsNum);
```

a)

Write the action that will return all the VALUE from the matrix Data.

4. Bug validation:

```
#starting testcopy
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0

Program received signal SIGSEGV, Segmentation fault.
0x00406361 in Matrix::copy (this=0x68f6dc) at Matrix.cpp:191
191 copy.setElement(matrixData[j][i], j, i);
(gdb)
```

a)

Still get sigsegv this is happening due to the wrong copy structure, miss up x and y.

5. Bug detection:

```

for(int i = 0; i < rowNum; i++)
    for(int j = 0; j < colsNum; j++)
        copy.setElement(matrixData[j][i], j, i);

return copy;

```

a)

Wrong dude, this method caused the matrix first ordered the cols then rows.

6. Bug fixing valid:

```

t
#starting testCopy
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0

#success testCopy OK

#starting testAddToInvalid

#failure testAddToInvalid Test.cpp:187 testAddToInvalid: expected == actual expected: Invalid operation.
Empty for Now

#starting testAddToValid

```

a)

Nice, the test copy successfully passes!