Lab 4 Debugger Report

- Bug observation
  - Function Matrix Matrix::copy() has semantic bugs which prevents the test case to run properly as the test cases were halted when testcopy() was called
  - ```
    #starting testCopy
    ```
  - The program is expected to create an instance of a matrix that's exactly the same as the matrix itself.
- GDB analysis
  - Breakpoint at line 209
    - ```
      (gdb) break Matrix.cpp:209
      Breakpoint 1 at 0x406251: file Matrix.cpp, line 209.
      ```
    - ```
      209     Matrix copy = Matrix();
      ```
  - Breakpoint at line 211
    - ```
      (gdb) break Matrix.cpp:211
      Breakpoint 2 at 0x40625b: file Matrix.cpp, line 211.
      ```
    - ```
      211     for(int i = 0; i < rowsNum; i++)
      ```
  - Breakpoint at line 213
    - ```
      (gdb) break Matrix.cpp:213
      Breakpoint 3 at 0x40627e: file Matrix.cpp, line 213.
      ```
    - ```
      213             copy.setElement(matrixData[j][i], j, i);
      ```
  - First semantic bug:
    - ```
      Matrix::Matrix (this=0x68f6d0) at Matrix.cpp:21
      21              rowsNum = 3;
      (gdb) s
      22              colsNum = 3;
      ```
    - The test case shows that the rowsNum and colsNum should 4 and 5 since that's the supplied matrix's dimension.
  - Second semantic bug:
    - ```
      (gdb) i locals
      j = 1
      i = 0
      ```
    - ```
      (gdb) i locals
      j = 2
      i = 0
      copy = {rowsNum = 3, colsNum = 3, matrixData = 0x11519e8}
      ```
    - Looking at the indices feeding into the setElement function, the indices are flipped
      - It should have been [0, 0] → [0, 1] , [0, 2]
      - However, current program shows it starts with [0,0] then goes to [1,0], and then [2,0]
- Possible root cause
  - For the first semantic bug
    - This is due to the fact that the original function used matrix() – the default constructor instead of the specialized constructor. The default constructor has a set dimension of [3][3] which mismatched with the desired matrix's dimension.

- For the second semantic bug
  - The row and column indices are flipped in the initial implementation, it should've been [row][col] instead of [col][row]

- Bug fix validation
  -
```
Matrix copy = Matrix(rowsNum,colsNum);

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

return copy;
```
  - The fix included using specialized constructor Matrix(int i, int j) instead of the default constructor in order to create the matrix with the desired dimension.
  - The second fix includes flipping the matrix indices to the proper order [row][col]
  -
```
#starting testCopy

#success testCopy OK
```
  - After the fix, the test case runs successfully