CS 170
Assignment 4
Dr Malek Mouhoub

Develop a C++ class to model the mathematical notation of a matrix of naturals. Your class should include the following functions.

- "+": Addition of 2 matrices.
- "-": Substration of 2 matrices.
- "*": Multiplication of 2 matrices.
- Scalar multiplication (multiply all the elements of a matrix by a given integer).
- Power. Raise a matrix to power n.
- "==" : returns true if both matrices are equal.

Note that, in the above, the operators "+, -, *, ==" need to be overloaded. Before performing the required data manipulation, each function you write should validate its input arguments. In particular, the dimensions of the matrices involved in an operation must be compatible for that operation. A simple run of the driver program follows.

```
Enter the number of rows of matrix 1: 2
Enter the number of columns of matrix 1: 2
Enter the elements of matrix 1 row by row:
1 0
0 1
Enter the number of rows of matrix 2: 2
Enter the number of columns of matrix 2: 2
Enter the elements of matrix 2 row by row:
1 1
1 0

matrix 1 + matrix 2:
2 1
1 1

matrix 2 - matrix 1:
0 1
1 -1

matrix 1 x matrix 2:
1 1
1 0

matrix 1 == matrix 2?
no

n x matrix 2. Enter the number n: 4
4 4
4 0

matrix 2 power n. Enter n: 2
2 1
1 1
```

Hand In

1. The header, implementation and driver program should be named: `MyMatrix.h, MyMatrix.cpp` and `TestMyMatrix.cpp`. Your C++ program **SHOULD** compile using `CC` (Sun compiler) under Hercules.
2. Submit all the above files using WebCT: `www.uregina.ca/webct`. You will then receive an acknowledgement email confirming your submission. You should save this email as a proof of submission. If you do not receive an email acknowledging your submission then you should promptly email the marker (mark115@cs.uregina.ca) with your submission in attachment.

Marking scheme: total = 100% + 10% (Bonus)

1. Readability (program style) : 10%
   - Program easy to read,
   - well commented,
   - good structured (layout, indentation, whitespace, ...) and designed(following the top-down approach)
2. Compiling and execution process : 10%
   - program compiles (with CC under hercules) w/o errors and warnings
   - robustness : execution w/o run time errors
3. Correctness : 80%
   - code produces correct results (output).
   - **output meets the initial requirements (see above for the output format)**.
4. Bonus : 10%
   - Features that increase functionality and/or presentation.