MF 703 – C++ for Mathematical Finance
Fall 2015
Homework Assignment 2

1. **(25 pts)** Rewrite the program you developed for Assignment 1 by defining the functions
   - mean
   - sampleStDev (the sample standard deviation)
   - min and max

   The input data for the functions is in the format

   mm/dd/yyyy, number

   where mm-month, dd-day, yyyy-year, and number some floating number, e.g.

   05/27/2004, 1121.28

   Write a main program (called driver) to test your program using the file SPXPricesFromCBOE.txt that contains daily prices of the S&P500 Index from 01/02/2004 to 12/31/2004 downloaded from http://www.cboe.com and provided in the assignment folder along with the assignment statement. (If interested you can find more historical data at http://www.cboe.com/micro/bxm/historical.aspx).

   To read the prices from the file include <fstream> and define an input file as in Assignment 1. To reach the value of the price (1121.28) one has to get rid of the 11 characters that precede it. Here are two options to do it:

   i. loop through the first 11 characters, reading and discarding one at a time:

```
char c;
for(int i=0; i < 11; i++)
       infile.get(c);
```

   ii. use the ignore(<n> , <delim>) function from <fstream> which takes two parameters:

   n -- specifies the number of characters to extract/ignore

   delim – character that specifies an end character for the sequence. For instance

```
infile.ignore(11,',');
//extracts/ignores up to 11 character or until ',' whichever comes first
```

2. **(25 pts)** ***Roots: The Newton-Raphson*** approximation finds the zeros of f(x) = 0 by computing

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}$$

until $|x_n - x_{n-1} < \varepsilon|$

where f'(x) is the derivative of f(x), and the ε some predefined error. To start the approximation one needs to choose an initial guess for x_o. This initial guess is important: an unlucky choice can lead to infinite numbers of iteration leading away from the solution.

Write a function root that uses the Newton-Raphson approximation to compute the n-th root of x, i.e. find the zeros of the function

$f(x) = x^n - a = 0$

(function root takes n as argument and returns the n-th root of x).

Write a program to test your function (such test program is referred to as driver or test harness). The driver should

- Tell the user what the function does
- Ask for input (here it needs to ask the user to input n and a)
- Validate the input by printing it on the screen
- Compute the n-th root and print it on the screen
- Ask the user if (s)he wants to continue and through the following dialogue:

Do you want to compute a root ? (-1 to exit): -1
<Termination:

i.e. the program terminates if the user enters '-1' and repeats the simulation otherwise.

3. **(25 pts)** Write a function **reverseDigits** that takes an integer value between 1 and 9999, and returns the number with its digits reversed, e.g. if given 3456 it returns 6543. Write a driver to test your program that allows the user to enter and test numbers.
The user should be able to repeat the calculation or enter -1 to exit as described in #2.

Extra credit Options: a) Generalize the program for any integer number. b) Give an iterative and a recursive solution.

4. **(25 pts)** ***Computing e^x through power series***. Write a function **exponential** that computes the value of e^x through the series:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

To achieve the desired precision keep adding terms until the absolute value from of the new term becomes less than a given threshold. For the absolute value you can use the function fabs() from the <cmath> library that returns the absolute value of its argument (more at http://www.cplusplus.com/reference/clibrary/cmath/fabs/). You can also use abs() but this function is available only in C++ for double argument; in C it is defined only for int and long arguments and is part of <cstdlib> not of <cmath>. (more on http://www.cplusplus.com/reference/clibrary/cmath/abs/ and http://www.cplusplus.com/reference/clibrary/cstdlib/abs/)
Write a driver to test your program that allows the user to enter the desired precision.
Provide the user with the ability to repeat the calculation or enter -1 to exit as described in #2.