Assignment 5

Submit 1,3 and 5:

1. Explore different ways to compute the two examples done in class: the Fibonacci sequence and the factorial of a number N:
   a. Using `for` loops and using `while` loops
   b. For factorial, computing from N downwards, rather than 1 upwards.
   c. For Fibonacci, what answers do you get when you print out 50 terms? Is there anything wrong? How can you fix it?

2. Write a program that reads in an integer, N, and computes the first N *Pell* numbers. The *Pell* sequence of integers is {0, 1, 2, 5, 12, 29, 70, 169, 408, 985, 2378 …} and is defined as
   P[0] = 0, P[1] = 1, P[N] = 2*P[N-1] + P[N-2]

3. Write a program that reads in an integer, N, and computes and displays the first N *Pavodan* numbers. The *Pavodan* sequence of integers is {1, 1, 1, 2, 2, 3, 4, 5, 7, 9, 12, 16, 21, 28, 37, 49, 65, 86, 114, 151, 200, 265,…} and is defined as
   P[1] = P[2] = P[3] = 1, P[N] = P[N-2] + P[N-3], N>=4.

4. Write a program that reads four floating point values `w,x,y,z` and modifies them so that `w` now contains the minimum value, `x`, contains the next smallest value, `y`, contains the next smallest value, and `z` contains the maximum value.

5. Write a program that declares a `float sum`, reads in an integer value `N` and computes and displays the sum of the inverse of the first N numbers i.e. computes
   `sum = 1 + 1/2 +1/3 + 1/4 + 1/5 + … + 1/N`.
   Try the program for very large values of N e.g. N=1000000000. Do you notice anything strange about the result? How about if you declare `double sum` i.e. use a double-precision floating-point value for the sum instead?