CS3383, Winter 2019 Assignment # 6
Rasoul Shahsavarifar

Faculty of Computer Science, UNB **Due time:** Friday, Mar/1/2019, 1:30 p.m

---

Student's full name: .... Student ID:..................

Note:

- No submission after the due time will be accepted.
- The full credit will be given only for correct solutions that are described clearly.

Question 1 (14 marks) (Based on exercise 6.1 of DPU textbook) A *substring* (or *continuous subsequence*) of a sequence S is a subsequence made up of consecutive positions of S. For example, if S is

$$5, 15, -30, 10, -5, 40, 10$$

then $15, -30, 10$ is a substring of S but $5, 15, 40$ is not.
Consider the problem of finding the substring of maximum sum:
Input: A sequence $a_1, a_2, \ldots, a_n$ of numbers.
Output: A substring of maximum sum.
Note that a substring of length 0 has sum 0.

a) Design and write (in pseudocode) a linear-time dynamic programming algorithm that solves this problem. (Note that this is to find such a substring, not just its sum.)

b) Implement your algorithm from (a) in $C++$, or $Java$. Hand in your code and I/O from at least three suitable test cases that demonstrate how well it handles various situations.

Question 2 (6 marks) Computing the number of combinations of size k of a set of n items $C(n,k) = \binom{n}{k}$, where $\binom{n}{k} = \frac{n!}{(n-k)!k!}$ on a computer can be awkward. Computing the numerator and denominator separately and then dividing tends to overflow the integer representation for the intermediate calculations; on the other hand, computing the overall result as the product of floating-point ratios $\frac{n}{k} \cdot \frac{n-1}{k-1} \cdot \ldots \cdot \frac{n-k+1}{1}$ can introduce rounding errors.
To get a correct integer answer without over owing, we can use Pascal's Formula, which defines the number of combinations using the following recurrence:

$$C(i,j) = \begin{cases} 1 & \text{if } j = 0 \text{ or } i = j \\ C(i-1, j-1) + C(i-1, j) & \text{if } 1 \leq j \leq i-1 \end{cases}$$

The recurrence is undefined if $j > i$, or if either i or j is negative.
Design and write a dynamic programming algorithm based on the above recurrence that will compute $\binom{n}{k}$ given n and k.