# CS 311 HW Assignment 2 (Due by 9am on Feb 13th)

## Programming 1 – Queue (20 points)

Write a C++ program to generate all strings using A, B, and C as the letters.

The strings must be generated in the following order:

A
B
C
AA
AB
AC
BA
BB
BC
CA
CB
CC
AAA
AAB
AAC
ABA
ABB
ABC
ACA
ACB
ACC
BAA
etc.

Implement the client program. The program should keep on going until your data structure overflows. At that point, **print out all the remaining** elements from the queue by using displayAll().

## **Bonus:** Programming 2 - Vector-based Stack (10 points as bonus)

Re-write the HW1 stack class (copy and then rename as vstack.h and vstack.cpp first) so that

- It now uses a vector instead of an array.
- There is no maximum size now. The stack starts out having no slots and it will grow as elements are added.
- No need for top. Take it out. You can call size()-1 to get top.
- Destructor has to do some work to make sure it leaves no cells behind.
- Constructor has no work to do.
- isFull() always returns false.
- Be careful that pop_back() and front() do not return the top item which is at size()-1.
- Client: in case of error, destructor will not be called but OK.

Now, use this stack class with the HW1 main.cpp program to evaluate post-fix expressions. Test with the same cases as before.

HW1 main.cpp should not be changed except to include the new header file. Your new implementation should be invisible to the client.

## Submission

SUBMIT ALL THE FOLLOWING FILES IN **A ZIP FILE** TO COUGAR COURSES:

Always make sure the files you submit can be compiled on empress.csusm.edu.

- queue.h -- Queue header file (for Problem 1)
- queue.cpp -- Queue source file (for Problem 1)
- client.cpp -- Problem 1 application file (for Problem 1)
- vstack.h -- Vector-based stack header file (for Bonus Problem 2)
- vstack.cpp -- Vector-based stack source file (for Bonus Problem 2)
- main.cpp -- Problem 2 application file (for Bonus Problem 2)

Note: Compress all the above files in a zip file, name it with your name, and submit the zip file on Cougar Course. For example, my first name is Xin and my last name is Ye, then I will name the zip file as XinYe.zip.

## Grading

1. On Cougar Course, submit all the files in a zip file with your name. Otherwise, we will not grade it.
2. Your code should be compiled on Cougar Course. If there is a compilation error, you will get 0 points.
3. If your code fails in 1 test case, we will deduct 10% of the total points.
4. If your code fails in 2 test cases, we will deduct 20% of the total points.
5. If your code fails in 3 test cases, we will deduct 40% of the total points.
6. If your code fails in more than 3 test cases, we will deduct 90% of the total points.
7. The comments in your code count for 10% of the total points.
8. Additionally, we will deduct 5 points for each day after the due date.