Name_____

1. Write a simple program to model a bank. The bank is to be implemented as a dynamic array of pointers to *CheckingAccount* structures. The bank must implement functions to open an account or close an account. As an account is opened or closed, the size of the array should expand or shrink. The bank should also support a function that will display the name and balance for each customer.
2. Using the operators new and delete, create a set of C++ functions for constructing dynamic strings. Memory for a dynamic string is to be allocated as necessary. Include functions to concatenate, append, and replace one string with a larger or smaller one. The function *concatenate()* combines two strings while *append ()* adds a character or substring to the end of a named string. For example,

   concat: S3 <- S1 + S2

   S1: "Bon Jour "

   S2: "mon petit frere"

   S3: "Bon Jour mon petit frere"

   S1, S2 remain unchanged

   append: S1 <- S1 + " Ca va, mon ami"

   S1: "Bon Jour Ca va, mon ami"

   replace: S1 <- S2

   S1: "mon petit frere"

   S2: "mon petit frere"

   Be certain to delete the dynamic instances before exiting your program.
3. Write a simple program that declares two integers, two pointers to point to those integers, and two pointers to pointers to refer to the pointers.
   a. Display the values of the integers directly, via the pointers, and via the pointers to pointers.
   b. Display the values of the pointers and the pointers to pointers. Explain your results and with a simple graphic, illustrate how these pointers may look in memory.

c. Write a function, swapPointers(int* ptr0, int* ptr1) that takes your two pointers and swaps them. Display the results following the swap in your function and in main. Explain any discrepancies.

d. Repeat part c with the modified prototype swapPointers(int** ptr0, int** ptr1). Once again, the intention is to swap the original pointers.