PROBLEM: **Assume that your computer has the very limited capability of being able to read and write only single-integer digits and to add two integers consisting of one decimal digit each. Write a program that can read two positive integers, add these integers together, and display the result.**

**Hint 1: Store the two numbers in two short arrays, one digit per array element. Use the global const int MAX\_DIGITS to specify the size of the arrays. You will need a loop to add the digits in corresponding array elements. Don’t forget to handle the carry digit if there is one!**

**Hint 2:**  **You read the list of digits into an array of MAX\_DIGITS. The first digit will be read into num[0] and the second digit num[1] and so on until you hit end of line. Make sure the total number of digits read in is less than or equal to MAX\_DIGITS. After you have read in the list you will have to "push" the list down so that the last digit is in num[MAX\_DIGITS-1] and the second to last digit is in num[MAX\_DIGITS-2] and so on. Afterward, leading digits not being used must be set to 0. You will have to do the same for the second number entered. After both values are read in and pushed down to the end of the array, you add the corresponding digits from both arrays, that is, adding num1[MAX\_DIGITS-1] to num2[MAX\_DIGITS-1], then adding num1[MAX\_DIGITS-2] to num2[MAX\_DIGITS-2] and so on until you add num1[0] to num2[0]. When adding 2 digits, if the sum is greater than 9, you will have to carry the one over to the sum of the next digits on the left.**

**Hint 3: Use code similar to line of code below to input a single digit as a char:**

char charDigit = cin.get();

**Hint 4: Use code similar to line of code below to convert an ASCII char '0' to '9' to the integer value it represents:**

unsigned short = charDigit - '0';

**Assume that charDigit contains an ASCII value between '0' and '9'. How does the above line of code work? charDigit will have an ASCII decimal value between 48 and 57 (look up ASCII table with Google to verify what I'm saying). The char '0' is represented by decimal value 48 , ... , char '9' is represented by decimal 57. By subtracting '0' from the ASCII value, we are in affect subtracting 48. The net result will be between 0 and 9, depending on the ASCII digit being represented by charDigit.**

**NOTE 1: Start with my supplied start file. I do not want you to change ANY of the code I have supplied in the supplied start file. I do not want you to add ANY additional functions.**

**NOTE 2: Your output should be IDENTICAL to the output shown in the 2 screenshot text files.**

**NOTE 3: Why did I make the arrays being passed type "short" ? The largest value that will be stored in an element of an array is 9. We only need 4 bits to represent an unsigned value of 9. A short is 16 bits long, clearly plenty large enough to represent the largest decimal digit. We could actually use arrays of type char instead, because a char is 8 bits and is again plenty large. As well, we can perform arithmetic operations with variables of type char. But the code would be a little trickier, so I have gone with the simpler solution of using arrays of type short.**

**NOTE 4: I have included in Start folder 2 .exes for you to experiment with where MAX\_DIGITS is set to 30.**