CMPT 130: Lab Work Week 7

Most of the questions below will ask you to write a function. It is your responsibility to create a suitable test main program for each function and test your functions for correctness. In your test main program, create suitable array or string, manually analyze it (them) and think what your function should perform or return; then run your program and see if your function does its job correctly.

In some of the questions, you may be asked to write a function that may require complicated loops or logical thinking. In such cases, it is advised you break down your solution to some simple and high level steps and then implement the steps as functions; so that the main program may call a function and then that function may call other supporting function(s) to make problem solving easier.

See the following complete example and use this idea as a blue print whenever you face problems that seem too complicated to be solved in one function.

Example Question: Write a C++ program that creates an array of integers of size 10, populates the array with random integers in the range [2, 100], and finally prints the elements of the array that are prime numbers.

Solution: Design the main program to create an array of integers of size 10 and populate it with random integers in the range [2, 100]. Then call a function to print the elements of the array that are prime numbers. This function should be designed to loop on the elements of the array and for each element of the array print it only if it is prime number as determined by yet another function that takes an integer and returns true or false depending if the integer number is a prime or not.

```
bool isPrime(const int x)
{
     //Given integer x, if any number between 2 and x-1 divides x then x is not a prime.
     //If no number betwen 2 and x-1 divides x then x is prime
     for (int i = 2; i < x; i++)
         if (x % i == 0)
             return false;
     return true;
}
void printPrimes(const int arr[], const int size)
{
     //Loop on the elements of the array and for each element check if it
     //is prime by calling another function that tests for prime.
     //If it is prime then print it. Otherwise don't print it.
     for (int k = 0; k < size; k++)
         if (isPrime(arr[k]) == true)
             cout << arr[k] << endl;
}
int main()
{
     //Step 1. Create an array
     int A[10];
     //Step 2. Populate the array with random integers in the range 2 to 100
     for (int i = 0; i < 10; i++)
         A[i] = rand() % 99 + 2;
     //Step 3. Print the elements of the array
     cout << "The elements of the array are..." << endl;
     for (int i = 0; i < 10; i++)
         cout << A[i] << endl;
     //Step 4. Print the elements of the array that are prime numbers
     cout << "The elements of the array that are prime numbers are..." << endl;
     printPrimes(A, 10);

     system("Pause");
     return 0;
}
```

Code Analysis Questions

1. Consider the following program and answer the following questions without typing the code onto a computer.
   - **a.** Does the program have any syntax error?
   - **b.** Does the program have any runtime error?
   - **c.** Does the program have any semantic error?
   - **d.** What is the output of the program?

   Once you have your answers ready, type the code EXACTLY AS IT IS onto a computer and run the program and see if you have got correct answers. If not re-do the question and analyze it until you figure out what is going in the program. Finally use the constant modifier on the function parameters and see how that helps to understand the program better.

```
int countEven(int arr[], int size)
{
    //Let's not use the modulo operator. Instead let's do the actual math for checking even.
    //Note that an integer value a is even if the value of a is equal to 2*(a/2)
    int count = 0;
    for (int i = 0; i < size; i++)
    {
           if (arr[i] = 2*(arr[i]/2))
                  count++;
    }
    return count;
}
int main()
{
    int A[5];
    //Initialize the elements of the array
    for (int i = 0; i < 5; i++)
           A[i] = i+1;
    //Print the elements of the array
    cout << "Originally the elements of the array are" << endl;
    for (int i = 0; i < 5; i++)
           cout << A[i] << endl;
    //Count and print how many even number elements there are in the array
    int m = countEven(A, 5);
    cout << "There are " << m << " even numbers in the array." << endl;
    //Print the elements of the array
    cout << "After the function call, the elements of the array are" << endl;
    for (int i = 0; i < 5; i++)
           cout << A[i] << endl;

    system("Pause");
    return 0;
}
```

Counting Problems

2. Write a program that creates a C++ static array of integers of length 10, populates the elements of the array with random integers and then prints how many of them are even integers and how many are odd.

3. Write a program that creates a C++ static array of integers of length 10, populates the elements of the array with random integers and then prints how many of them are prime integers and how many are not prime integers.

4. Write a program that creates a C++ static array of floats of length 10, populates the elements of the array with random floats in the range -1.0 and 1.0 and then prints how many of them are positive floats and how many are negative floats.

5. Write a function named **evenCounter** that takes an array of integers and its size as arguments and returns the number of even elements in the array.

6. Write a function named **primeCounter** that takes an array of integers and its size as arguments and returns the number of elements of the array that are prime numbers. Assume the array contains positive integers greater than 1.

7. Write a function named **vowelCounter** that takes an array of characters and its size and returns the number of vowel characters in the array. Vowels are 'a', 'e', 'i', 'o', 'u', 'A', 'E', 'I', 'O' and 'U'

8. Write a C++ function named **countElements** that takes three arguments: an array of integers, its size, and an integer and that returns the number of elements in the array that are equal to the integer argument.

9. Write a C++ function named **countElementsFromIndex** that takes four arguments: an array of integers, its size, an integer value x, and an integer index. Your function must search the integer value x in the array starting from the index argument upto the last element of the array and return the number of elements in the array that are equal to the integer argument. For example give the following code

```
const int size = 11;
int A[]= {5, 7, 8, 0, 5, 4, 1, 5, 6, 9, 5};
int searchValue = 5;
int startIndex = 2;
```

Then the function call **countElementsFromIndex(A, size, searchValue, startIndex);** must return 3 because there are 3 elements equal to 5 starting from index 2. Of course in total there are 4 elements equal to 5 but because the search starts at the index given by the index argument, then the function will find only 3 elements equal to 5.

10. Write a C++ function named **countCommonElements** that takes four arguments: an array of integers, its size, a second array of integers, and its size. Your function must return the number of elements of the first array that are also found in the second array. Assume each of the arrays contains distinct (i.e. different) elements.

11. Write a function named **distinctElementsArray** that takes an array of integers and its size as arguments and returns true if the array contains distinct (i.e. different) elements; otherwise returns false. What is the pre-condition for the function?

Squeezing Problems

12. Write a program that creates a C++ static array of integers of length 10, populates the elements of the array with random integers in the range [-10, 10] and then prints the sum of all the elements.

13. Write a program that creates a C++ static array of integers of length 10, populates the elements of the array with random integers in the range [-10, 10] and then prints the sum of the absolute values of all the elements.

14. Write a program that creates a C++ static array of integers of length 10, populates the elements of the array with random integers in the range [-10, 10] and then prints the absolute value of the sum of all the elements.

15. Write a program that creates a C++ static array of integers of length 10, populates the elements of the array with random integers in the range [-10, 10] and then prints the product of all the elements.

16. Write a program that creates a C++ static array of integers of length 10, populates the elements of the array with random integers in the range [-10, 10] and then prints the product of the absolute values of all the elements.

17. Write a program that creates a C++ static array of integers of length 10, populates the elements of the array with random integers in the range [-10, 10] and then prints the absolute value of the product of all the elements.

18. Write a program that creates a C++ static array of integers of length 10, populates the elements of the array with random integers in the range [-10, 10] and then prints the maximum and minimum elements of the array.

19. Write a C++ function named **maxElement** that takes an array of integers and its size as arguments and returns the maximum element. What is the pre-condition for the function?

20. Write a C++ function named **minElement** that takes an array of integers and its size as arguments and returns the minimum element. What is the pre-condition for the function?

21. Write a C++ function named **sumArray** that takes an array of floats and its size as arguments and returns the sum of the elements in the array.

22. Write a C++ function named **productArray** that takes an array of floats and its size as arguments and returns the product of the elements in the array.

Existence of specific element Problems

23. Write a C++ function named **isFound** that takes three arguments: an array of integers, its size, and an integer and that returns true if the integer argument is found in the array; otherwise return false.

24. Write a C++ function named **isFoundFromIndex** that takes four arguments: an array of integers, its size, an integer x, and an integer index. Your function must return true if the x is found inside the array starting from the index argument up to the last element of the array; otherwise return false. What is the pre-condition for the function?

25. Write a function named **containsEven** that takes an array of integers and its size; and that returns true if the array contains at least one even number element otherwise return false.

26. Write a function named **containsDigit** that takes an array of characters and its size; and that returns true if the array contains a digit otherwise return false.

27. Write a function named **containsVowel** that takes an array of characters and its size; and that returns true if the array contains a vowel otherwise return false.

28. Write a function named **containsLowerCase** that takes an array of characters and its size; and that returns true if the array contains a lower case English letter otherwise return false.

29. Write a function named **containsUpperCase** that takes an array of characters and its size; and that returns true if the array contains an upper case English letter otherwise return false.

30. Write a function named **isAlpha** that takes an array of characters and its size; and that returns true if the array contains only English alphabets; otherwise return false.

31. Write a function named **isAlphaNumeric** that takes an array of characters and its size; and that returns true if the array contains only English alphabets and digits; otherwise return false.

32. Write a function named **containsPrime** that takes an array of integers and its size; and that returns true if the array contains at least one prime number element otherwise return false.

33. Write a function named **containsComposite** that takes an array of integers and its size; and that returns true if the array contains at least one composite number element otherwise return false.

34. Write a program that creates a C++ static array of integers of length 10, populates the elements of the array with random integers in the range [-10, 10] and then prints the message **"a negative number found in the array"** if the array contains at least one negative integer and prints the message **"No negative number found in the array"** if the array does not contain any negative number.

35. Write a program that creates a C++ static array of integers of length 10, populates the elements of the array with random integers and then prints the message **"a prime number found in the array"** if the array contains at least one prime number and prints the message **"No prime number found in the array"** if the array does not contain any prime number.

36. Write a program that creates a C++ static array of characters of length 10, populates the elements of the array with random characters and then prints the message **"a digit found in the array"** if the array contains at least one digit and prints the message **"No digit found in the array"** if the array does not contain any digit. For random characters consider only those characters whose ASCII code is in the range [33, 126] in decimal.

37. Write a program that creates a C++ static array of characters of length 10, populates the elements of the array with random characters and then prints the message **"an alphabet found in the array"** if the array contains at least one alphabet and prints the message **"No alphabet found in the array"** if the array does not contain any alphabet. For random characters consider only those characters whose ASCII code is in the range [33, 126] in decimal.

38. Write a function named **isIncreasing** that takes an array of integers and its size and that returns true if the elements of the array are in increasing order; otherwise return false. What is the pre-condition for the function?

Miscellaneous Problems

39. Write a function that takes an array of integers and its size as arguments and that rearranges the array such that the even integer elements of the array are placed at the beginning of the array and the odd integer elements of the array are placed at the end of the array. A sample test program and its output is given below to help you understand the question better.

```
int main()
{
	//Define an array
	const int SIZE = 20;
	int A[SIZE];

	//Populate the array with some integers in the range [0, 29]
	for (int i = 0; i < SIZE; i++)
		A[i] = rand() % 30;

	//Print the elements of the array
	cout << "Originally the elements of the array are" << endl;
	for (int i = 0; i < SIZE; i++)
		cout << A[i] << "   ";
	cout << endl;

	//Re-arrange the array so that even integers are at the beginning
	reArrangeArray(A, SIZE);

	//Print the elements of the array
	cout << "After re-arranging the elements of the array are" << endl;
	for (int i = 0; i < SIZE; i++)
		cout << A[i] << "   ";
	cout << endl;

	system("Pause");
	return 0;
}
```

OUTPUT

```
Originally the elements of the array are
11    17    4    10    29    4    18    18    22    14    5    5    1    27
1    11    25    2    27    6
After re-arranging the elements of the array are
6    2    14    22    18    18    4    10    4    11    17    29    5    5
1    27    1    11    25    27
Press any key to continue . . .
```

Please note that the order of the elements within the same group is not important (i.e. it is irrelevant) which means your order of elements within each group can be different and that is okay.

40. Write a function that takes an array of characters and its size as arguments and that rearranges the array such that the lower case character elements of the array are placed at the beginning of the array and the upper case character elements of the array are placed at the end of the array. Assume the array argument contains only English alphabet characters. The order within the same group is irrelevant.

41. Write a function that takes an array of non-negative integers and its size as arguments and that rearranges the array such that the prime number elements of the array are placed at the beginning of the array and the non-prime number elements of the array are placed at the end of the array. Assume each element of the array is greater than 1. The order within the same group is irrelevant.

C++ Strings

42. Write a function named **randomName** that takes no argument and returns a random name string of 10 characters made up of only English alphabets. The first character of the name must be uppercase while all the remaining characters must be lowercase.

43. Write a C++ function named **isFound** that takes two arguments: a string and a character. Your function must return true if the character argument is found in the string; otherwise it must return false.

44. Write a function named **vowelCounter** that takes a C++ string and returns the number of vowel characters in the C++ string.

45. Write a C++ function named **countCharacter** that takes a string and a character as arguments and returns the number of times the character is found in the string.

46. Write a C++ function named **isDistinct** that takes a string argument and returns true if the string argument contains distinct characters otherwise returns false. What is the pre-condition for the function?

47. Write a C++ function named **countCharacterFromIndex** that takes a string, a character and an integer index as arguments and returns the number of times the character is found in the string starting from the given index argument. What is the pre-condition for the function?

48. Write a C++ function named **countCommonChars** that takes two string arguments s1 and s2 and then returns the number of characters of s1 found in s2. For simplicity you can assume the string s1 contains distinct characters and also the string s2 contains distinct characters.

49. Write a C++ function named **commonCharsString** that takes two string arguments and returns a new string made up of all the characters of s1 that are found in s2. For simplicity you can assume the string s1 contains distinct characters and also the string s2 contains distinct characters.

50. Write a program that declares an array of string data type with size 5, initializes each element of the array with a random name by calling your **randomName()** function defined in Q42, prints each element of the array, and finally prints the name that comes first in increasing order of the string elements of the array and also the name that comes last in increasing order of the string elements of the array.

51. Write a function named **increasingOrderStrings** that takes an array of string data type and its size and then returns true if the elements are in increasing order otherwise returns false. What is the pre-condition for the function?