CS&E 1222

Lab 10 – Array Pointers Lab Assignment – 20 points

- ✓ The *lab* must be accomplished solely by you:
  - ➢ DO NOT look at anyone's code other than your own, including code from another's student in your section or another section of the course, or any third party source, e.g. the Internet
  - ➢ DO NOT share or copy anyone else's code for any graded assignment
  - ➢ DO NOT work in pairs or groups
- ✓ All cases of academic misconduct will be reported to the *Committee On Academic Misconduct* (COAM).

Setting up the Programming Environment

Effective commenting and tabbing will affect your grade. The "style" of your program should follow the style of the sample programs in the lecture notes. Also see the example code from Lab #1. Your program should have the file name, your name, creation and last modification dates and a brief description of the program in the comments. ***In addition, read the document on "Commenting" found in the Content tab on Carmen under "Tutorials"***.

1. At the Linux command line type `mkdir lab10`. This will create a new directory named **`lab10`**. Work out of this directory. In order to do that, type `cd lab10`. This changes the current working directory to the directory **`lab10`**.
2. If you have created the directory **`lab10`**, then just type `cd lab10`.
3. Copy the file **`split_solution.exe`** from the directory **`/class/cse1222/9643/lab10`** into the current directory.

```
cp /class/cse1222/9643/lab10/split_solution.exe .
```

Be sure to include ***9643*** (this is your course section indicator) and the period, ".".

Programming Assignment

Write a program in a new file called **`split.cpp`** which reads a list of numbers into an array, splits the list into a list of negative numbers and a list of non-negative numbers, each in their separate arrays, and then outputs the arrays of negative and non-negative numbers. A *negative integer* is one that is strictly less than 0. All other integers are *non-negative integers*. The program should use the command **`new`** to allocate arrays of exactly the correct size.

The program should have a function count() for counting the number of negative elements and the number of non-negative elements of an array, a function split() for splitting the list into negative and non-negative lists, and a function print_array() for printing an array of numbers.

Run **`split_solution.exe`** to see an example of the program.

1. All functions should be written AFTER the main procedure.
2. A function prototype should be written for each function and placed BEFORE the main procedure.
3. Each function should have a comment explaining what it does.
4. Each function parameter should have a comment explaining the parameter.
5. Prompt for the number of elements of the list.

6. Allocate an array whose size equals the number of elements specified by the user.

7. Read into the array the elements of the list.

8. Write a function count() which counts the number of negative elements and the number of non-negative elements of an array. The function should take four parameters, the array, the number of elements in the array, the number of negative elements and the number of non-negative elements.

   The function sets the number of negative and non-negative elements but does not return any value.

9. In the main program, call the function count() to get the number of negative elements and the number of non-negative elements of the array.

10. Allocate an array whose size equals the number of negative elements.

11. Allocate an array whose size equals the number of non-negative elements.

12. Write a function split() which takes as input 3 arrays, *A*, *B*, and *C*, and stores the negative elements of *A* in *B* and the non-negative elements of *A* in *C*. (Use better array names than *A*, *B*, and *C*). The size of array *B* should be exactly the number of negative elements in array *A*. The size of array *C* should be exactly the number of non-negative elements in *A*.

    As you copy elements from *A* to *B* or *C*, count the number of elements copied to each array. When the function completes, check that the number copied equals exactly the array size. If it does not, print an error message and exit.

    The function should take six parameters, array *A*, the size of array *A*, array *B*, the size of array *B*, array *C*, and the size of array *C*.

    The function modifies arrays B and C but does not return any value.

13. Write a function print_array() which prints the elements of an array.

    The function should take two parameters, the array and the array length.

    The function does not modify or return any values.

14. 14. In the main program, call the function split(), passing the input array and the arrays whose sizes equal the number of even and odd elements.

15. In the main program, print the phrase “Negative elements: “. Print the array of negative elements using the function print_array().

16. In the main program, print the phrase “Non-negative elements: “. Print the array of negative elements using the function print_array().

17. Free the three dynamically allocated arrays. (Points will be deducted if you do not free the arrays).

18. Be sure to add the header comments “File”, “Created by”, “Creation Date” and “Synopsis” at the top of the file. Each synopsis should contain a brief description of what the program does.
19. Be sure that there is a comment documenting each variable.
20. Be sure that your if statements, for and while loops and blocks are properly indented;
21. Check your output against the output from the solution executables provided.
    Note: This lab requires you to dynamically allocate arrays of appropriate size. A large number of points will be deducted if you do not use dynamically allocated arrays, or do not allocate arrays of the correct size.

Submit Your Work

Important: Any program which does not compile and run will receive no credit!
If you are not sure what this means please ask your instructor.

Submit the file **`split.cpp`** using the *Lab 10* drop box on Carmen. **DO NOT** submit the file **`a.out`**. **DO NOT** submit uncompleted work from *Quiz 10*. This will not be graded.