**Lab 1: Due Wednesday April 9, 2014 11:59PM**

**Problem #1:**

Show the unix/linux commands that you would use to create a list of the smallest 5 files in a directory, sorted by decreasing file size.  
  
ll -S | grep ^- | tail -5  
  
  
**Problem #2:**  
  
Show the unix/linux commands that you would use to count the number of words in a text file which contains all of the letters a, b, c, d, e and f. These letters may occur more than once in the word and the word may contain other letters as well. (For example, "feedback" should be counted.)   
  
grep -o '\w\*a\w\*' file | grep b | grep c | grep d | grep e | grep f | wc -w  
  
**Problem #3:**

Show the unix/linux commands that you would use to create a 7x7 matrix of alternating entries of 1's and 0's. It should look like this:

1 0 1 0 1 0 1   
0 1 0 1 0 1 0   
1 0 1 0 1 0 1   
0 1 0 1 0 1 0   
1 0 1 0 1 0 1   
0 1 0 1 0 1 0   
1 0 1 0 1 0 1

yes 1 0 | fmt -w 14 | head -7  
  
**Problem #4 :**

Show the unix/linux commands that you would use to get a listing of all processes that you are currently running on the your Linux machine you are using, sorted by the command name in reverse alphabetical order (i.e. a process running zwgc should be listed before a process running acroread). The output should consist only of the processes you are running, and nothing else (i.e. if you are running 6 processes, the output should only have 6 lines).   
  
Linux Reference Commands: <http://www.mediacollege.com/linux/command/linux-command.html>

ps -ef | grep '^username' | sort -rk8  
  
**Problem #5 :**  
  
Write C++ function for binary Search   
**bool binarySearch(int\* anArray, int start, int end, int key);**   
<http://en.wikipedia.org/wiki/Binary_search_algorithm>

bool binarySearch(int\* anArray, int start, int end, int key)

{

if (end < start)

{

return false;

}

else

{

int middle = (start + end)/2;

if (anArray[middle] > key)

{

return binarySearch(anArray, start, middle-1, key);

}

else if (anArray[middle] < key)

{

return binarySearch(anArray, middle+1, end, key);

}

else

{

return true;

}

}

}  
  
**Problem #6 :**

Write C++ function for Bubble Sort Algorithm:

**void bubbleSort(int\* anArray, int start, int end);**

<http://en.wikipedia.org/wiki/Bubble_sort>

void bubbleSort(int\* anArray, int start, int end)

{

int temp;

for (int i = 0; i < end; i++)

{

for (int j = 0; j < end - i - 1; j++)

{

if (anArray[j] > anArray[j + 1])

{

temp = anArray[j];

anArray[j] = anArray[j + 1];

anArray[j + 1] = temp;

}

}

}

}  
  
C++ Reference Guide:

<http://www.cplusplus.com/>

**Attention!**

**Please upload your written solution on EEE , Lab 1 DropBox before the deadline, using .pdf. file format.**

**Please write your NAME and STUDENTID in the first line of the solution file.**

**Hints of Linux commands that you may use!**

**1) ps**

**2) tail**

**3) sort**

**4) cat**

**5) grep**

**6) head**

**7) yes**

**8) fmt**