EECE 230 Introduction to Programming, Sections 3 and 4
Quiz II

Dec 21, 2010

- The duration of this exam is 2 hours and 45 minutes.
- It consists of 4 problems.
- The exam is open book. You can use also all the material on Moodle: lecture notes, programming assignments, and solutions, etc. You are **NO**T allowed to use the **web** (**imail** included). You are not allowed to use **USB's** or files previously stored in your **account**.
- If you violate the above rules or if you communicate with a person other than the exam proctors during the exam, you will immediately get zero and you will be referred to the appropriate disciplinary committee.
- Active cell phones and any other unauthorized electronic devices are absolutely not allowed in the exam rooms. They should be turned off and put away.
- Plan your time wisely. Do not spend too much time on any one problem. Read through all of them first and attack them in the order that allows you to make the most progress.
- Submit your solutions each part in a separate file as indicated in the booklet. Include your name and ID number in each file. Submit the files online in a single zip file called *yourLastName.yourFirstName.zip*.
- Good luck!

Problem 1 (30 points). Files

a) **(15 points) Number of sentences**

Write a program which prompts the user to enter a C-string containing the name of an input text file. Your program is supposed to count the number of sentences in the file. Assume that each sentence ends with a period (dot). To simplify the problem, assume that periods are only used in the file to indicate the end of sentences.

For instance, the file consisting of

> *Write a program which prompts the user to enter a C-string containing the name of an input text file. Your program is supposed to count the number of sentences in the file. Assume that each sentence ends with a period (dot). To simplify the problem, assume that periods are only used in the file to indicate the end of sentences.*

contains 4 sentences.

Submit your code in a file called Prob1a.cpp including your name and ID number.

b) **(15 points) Length of the longest sentence**

Write a program which prompts the user to enter a C-string containing the name of an input text file. Your program is supposed to find the length of the longest sentence in the file. Assume as above that each sentence ends with a period and that periods are only used in the file to indicate the end of sentences. The length of a sentence is its number of characters including whitespaces and the terminating period.

For example, the length of longest sentence in the file example in Part (a) is 108 (the last sentence is the longest).

Submit your code in a file called Prob1b.cpp including your name and ID number.

Problem 2 (20 points). String mirror

Write a function *mirror*, which given two *C*-strings $s1$ and $s2$ as input argument, stores in $s2$ the string $s1$ followed by its reverse but without duplicating the last character. Assume that enough memory is allocated to $s2$ before calling the function.

Examples:

- If $s1$ = "Problem", then after calling the function *mirror* on $s1$ and $s2$, $s2$ becomes "ProblemelborP".
- If $s1$ = "ab", then after calling the function *mirror* on $s1$ and $s2$, $s2$ becomes "aba".
- If $s1$ is the empty string (i.e., consists only of the null character), then after calling the function *mirror* on $s1$ and $s2$, $s2$ becomes the empty string.

Test your function on the above examples.
Submit your code in a file called Prob2.cpp including your name and ID number.

Problem 3 (25 points). Recursive merge function

Recall the merge function we did in class. It takes two *sorted* arrays in nondecreasing order and it combines them into a single sorted array. The code on the merge function is in the slides on moodle.

In this problem you are asked to write a recursive version of the merge function. Call your function *recursiveMege*. The use of **for** or **while** loops in recursiveMege is strictly **prohibited**. You are asked to use recursion instead. A solution based on for or while loops is worth zero points.

Use the following test program.

```
#include <iostream>
using namespace std;

void printArray(const int A[],int n)
{
    for(int i = 0;i<n;i++)
    cout<<A[i]<<" ";
    cout<<endl;
}


void recursiveMerge(... parameters ...)
{
    ....
    body of the function
    ....
}

void main ()
{
   int L[] = {1,4,5};
   int R[]={2,3,7,10};
   int C[7];
   recursiveMerge(....);
   printArray(C,7);


   int L2[] = {1,4,5};
   int R2[]={10,20,21};
   int C2[6];
   recursiveMerge(...);
   printArray(C2,6);

   int L3[] = {10,40};
   int R3[]={1};
   int C3[6];
   recursiveMerge(....);
   printArray(C3,3);

}
```

You should get

```
1 2 3 4 5 7 10
1 4 5 10 20 21
1 10 40
```

Note that this problem is about the merge function and not about mergeSort (which is not included in Quiz II material).

Submit your code in a file called Prob3.cpp including your name and ID number.

Problem 4 (25 points). Three consecutive numbers

Write a function which given an array A of integers and its length n, checks whether or not A contains 3 consecutive integers (not necessarily in order or contiguous). Your function is only supposed to give a YES/NO answer.

Examples:

- The array $\{1, 110, 50, 6, 20, 4, 11, 13, 5, 23\}$ contains 3 consecutive integers $(4, 5, 6)$.
- The array $\{1, 110, 50, 20, 4, 11, 13, 5, 23\}$ does not contain 3 consecutive integers.
- The array $\{10, 11, 9\}$ contains 3 consecutive integers $(9, 10, 11)$.

Test your function on the above examples.
Submit your solution in a file called Prob4.cpp including your name and ID number.