Miguel Lopez 3/8/21

EE 5103 Mid-Term Exam - Spring 2021

Points: 100.

Submission deadline: Mar 08 by 11:59 PM.

You are allowed to consult whatever sources you wish for this exam for the purpose of learning concepts. However, do not use code that is not yours. That is, you should answer each of these questions on your own.

EXTRA CREDIT: If you follow the below submission instructions, you will get an extra 2 points in this exam.

Submission instructions: Name your pdf file for your solutions to questions 1-5 as ⟨LastNameFirstName⟩.pdf. Name your solution to question 6 as ⟨LastNameFirstName⟩.cpp. Place them both inside a folder called ⟨LastNameFirstName⟩. Zip this folder to obtain a file called ⟨LastNameFirstName⟩.zip. Upload this file by the deadline.

FULL NAME:

DO NOT USE VECTORS UNLESS STATED TO DO SO IN THE QUESTION.

1. (10 points) Implement a function that takes a temperature value in Centigrade and returns the corresponding value in Fahrenheit. Formula: °C x 9/5 + 32 = °F. For example: 37°C x 9/5 + 32 = 98.6°F.

```
double convert(double num)
{
  num= (num * 9/5)+32;
  return num;
}
```

2. (10 points) What is the output of the following program? If the output cannot be predicted, say "Cannot predict". Say "Compiler error" in case the program will not compile. Give a brief reason for your answer.

```
#include <iostream>
using namespace std;
int plusone(int);

int main(){
   int i=1;
   cout << i << endl;
   cout << plusone(i) << endl;
   cout << i << endl;
   return 0;
}

int plusone(int i){
    i=i+1;
    return i;
}
```

1
2
1

is the output.

- 'i' is initialized to '1'.
- plusone(int i) function added 1 to this => '2'!
- i value inside main was not changed => '1'.

(endl -> new line.)

13

3. (10 points) Write a function called `read_floats(int n)` that reads `n` `float` values from the user into an array and return the array back to the caller. Recall that a regular array will get deallocated when the function returns.

```
float* read_floats(int n)
{
   float *q= new float[n];
   for(int i=0; i<n; i++)
   {
     cin>>q[i];
   }
   return q;
}
```

4. (10 points) Write a function called `is_valid_phone_number` that takes a phone number as an array of characters and its size, and returns true if the input array is a valid phone number, false otherwise. A phone number is valid *only* if it is of the following format: (uuu) uuu-uuuu where u is from '0', '1', '2', etc. The parentheses, hypen and the space after ')' are necessary. Do not use the string or regex libraries.

```
bool is_valid_phone_number(char *phone, int size)
{
  if (size==14 && phone[0]=='(' &&
      isdigit(phone[1]) && isdigit(phone[2]) && isdigit(phone[3]) &&
      phone[4]==')' && phone[5]==' ' &&
      isdigit(phone[6]) && isdigit(phone[7]) && isdigit(phone[8]) &&
      phone[9]=='-' && isdigit(phone[10]) &&
      isdigit(phone[11]) && isdigit(phone[12]) && isdigit(phone[13]))
  {
     return true;
  }
  else
     return false;
}
```

5. (10 points) Write a function called matches that takes two int arrays and their respective sizes, and returns the number of consecutive values that match between the two arrays starting at index 0. Suppose the two arrays are $\{3, 2, 5, 6, 1, 3\}$ and $\{3, 2, 5, 2, 6, 1, 3\}$ then the function should return 3 since the consecutive matches are for values 3, 2, and 5.

```
int matches(int arr1[], int arr2[], int size1, int size2)
{
    int count=0, size=0;
    if(size1>size2)
      size=size1;
    else
      size=size2;
    for(int i=0; i<size; i++)
    {
      if(arr1[i]==arr2[i])
        count++;
      else
        return count;
    }
}
```

6. (50 points) This is the programming part of your exam. Download the file from Blackboard. Name your file as (LastNameFirstName).cpp. Upload your code back to Blackboard by the deadline.