Introduction to Computing
CS 151 - ON60
Department of Physics and Computer Science
Medgar Evers College
Exam 3

Instructions:

- **The exam requires writing a complete cpp file within an hour and 50 minutes. It requires completing tasks in four sections.**

- **Accompanying this file is a template cpp file. You must modify the cpp file; however, you cannot add additional libraries to or remove any libraries from the file. All other modifications are allowed.**

- **Tables can be constructed using a spreadsheet application instead of being included the the cpp file.**

- **Your submissions must be submitted to the Exams directory of your github repository and/or as attachments on Google classroom under the Exam03 assessment. The files must have the accurate extensions.**

- **Cheating of any kind is prohibited and will not be tolerated.**

- **Violating and/or failing to follow any of the rules will result in an automatic zero (0) for the exam.**

TO ACKNOWLEDGE THAT YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS ABOVE, AT THE BEGINNING OF YOUR SUBMISSION(S), ADD A COMMENT THAT CONSISTS OF YOUR NAME AND THE DATE

Grading:

Section	**Maximum Points**	**Points Earned**
Fundamentals	5	
Problem Solving	5	
Tracing	5	
Debugging	5	
Total	20	

Fundamentals

1. **In the commented section titled Fundamentals, for each of the following questions, write ONLY what is requested**

   a. Write the definition of a double function named `M()` that takes four double parameters and returns the average of its parameters.

   b. Write a statement that initializes a bool array to the solutions of the following true/false questions

   - a readonly pointer must be initialized when declared.
   - a pointer cannot be assigned a constant variable address.
   - a constant pointer must be initialized when declared.

   c. Write a statement that declares a dynamic double array with a size of 512.

   d. Given that a short array *t* with a size of 20 has been declared, write a statement(s) that assigns 0 to all the elements of *t*.

   e. Write the definition of a bool function named `S()` that takes two double parameters and a double reference parameter respectively. It assigns the lesser of the first two parameters to the reference parameter; and then, it returns true if the first parameter is assigned to the reference parameter, or returns false if the second parameter is assigned to the reference parameter.

Problem Solving

2. **Given a sequence of numbers, its running sum is a sequence such that the value of each element of the sequence is the sum of the values of the elements of the original sequence whose position is less than or equal to it. That is, if $a_1, a_2, \ldots, a_n$ is a sequence, the sequence $b_1, b_2, \ldots, b_n$ is its running sum if each element, b_k, is equal to**

$$b_k = \sum_{i=1}^{k} a_i = a_1 + \ldots + a_k$$

For instance, if you have the sequence $\{3, 4, 5, 9\}$, then its running sum would be $\{3, 7, 12, 21\}$.

Using the above information, write a double function named `SequenceData()` that takes double array parameter and an int parameter respectively. Given that the int parameter represents the size of the array parameter, the function makes the array its own running sum and returns the minimum value of the original values of the array. For instance, the caller `SequenceData(data,6)` where `data` equals `{2,6,5,1,8,3}` will return `1` and `data` will be `{2,8,13,14,22,25}` after the call.

Tracing

3. **In the commented section titled Tracing, construct a trace table (or list) of the caller `D("mOV3S")` where the definition of `D()` is below. The function `isalpha()` returns true if the character parameter is a letter.**

```
string D(string str)
{
  string a = "", b = "";

  for(int i = 0;str[i] != '\0';i += 1)
  {
    if(isalpha(str[i]))
    {
      a += tolower(str[i]);
    }
    else
    {
      b += str[i];
    }
  }
  return (a + b);
}
```

Debugging

4. **In the commented section titled Debugging, for each code segment, write ONLY the line number and the entire corrected line for each line that contains a syntax error and/or does not maintain the intent of the code.**

   a. /*Intent: given that the int parameter represents the size of the array parameter, it assigns consecutive integers to the elements of the array*/

```
01 | void I(string a[],int n)
02 | {
03 |   for(int i = 0;i <= n;i += 1)
04 |   {
05 |     a[i] = i + 1;
06 |   }
07 |
```

   b. /*Intent: returns the string "positive", "zero", or "negative" if the parameter is positive, zero or negative respectively*/

```
01 | string N(int n)
02 | {
03 |   if(n < 0)
04 |   {
05 |     return "positive";
06 |   }
07 |   else if(n > 0)
08 |   {
09 |     return "negative";
10 |   }
11 |   else(n == 0)
12 |   {
13 |     return 'zero';
14 |   }
15 | }
```

   c. /*Intent: given that the int parameter represents the size of the array parameter, it reads in values into the array parameter*/

```
01 | void G(int* c,int x)
02 | {
03 |   int t;
04 |
05 |   for(int i = 0;i < n;i += 1)
06 |   {
07 |     cin << t;
08 |     c[i] = t;
09 |   }
10 | }
```

   d. /*Intent: it returns true if none of its parameters are equal*/

```
01 | bool V(char a,char b,char c)
02 | {
03 |   bool A = (a != b && a != c && b != c)
04 |   return A;
05 | }
```