Introduction to Computing
CS 151 - ON60
Department of Physics and Computer Science
Medgar Evers College
Exam 2

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 Exam02 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. When would you use an if-else statement rather than simple if statement?

   b. Write a statement(s) that repeats the message `"Forever More"` indefinitely.

   c. Write the function prototype of a void function named `T()` that takes an int parameter and two string reference parameters respectively.

   d. Given that an int variable *s* has been declared, write a statement(s) that assigns the sum of the odd multiples of 7 between 1 and 500 inclusively to *s*.

   e. Write the definition of an int function named `A()` that takes an int parameter and returns twice the square of the parameters.

Problem Solving

2. **A system of linear equations with two unknowns is of the form**

$$\begin{array}{rcrcl} \mathbf{a_1 \cdot x} & + & \mathbf{a_2 \cdot y} & = & \mathbf{s_1} \\ \mathbf{a_3 \cdot x} & + & \mathbf{a_4 \cdot y} & = & \mathbf{s_2} \end{array}$$

where a_1, a_2, a_3, a_4, s_1 and s_2 are real numbers, and x and y are real number variables. A solution of a system of linear equations is an ordered pair (c, d) such that when c and d are substituted for x and y respectively in both equations, both equations will evaluate correctly. A system of equations can have either exactly one solution, infinite solutions or no solutions; which means, it is *independent*, *dependent* or *inconsistent* respectively.

The amount of solutions of a system of linear equations is can be determined using the function

$$\textbf{number of solutions} = \begin{cases} 1 & \textbf{if } det \neq \mathbf{0} \\ \infty & \textbf{if } det = \mathbf{0} \textbf{ and } \mathbf{a_1 = n \cdot a_3, a_2 = n \cdot a_4, s_1 = n \cdot s_2} \\ 0 & \textbf{otherwise} \end{cases}$$

where n is a real number and *det* is called the *determinant*, which is calculated as follows

$$det = \mathbf{a_1 \cdot a_4 - a_2 \cdot a_3}$$

Furthermore, if the system of equations is independent, its solution is in the form

$$\left(\frac{\mathbf{s_1 \cdot a_4 - s_2 \cdot a_2}}{\mathbf{a_1 \cdot a_4 - a_2 \cdot a_3}}, \frac{\mathbf{s_2 \cdot a_1 - s_1 \cdot a_3}}{\mathbf{a_1 \cdot a_4 - a_2 \cdot a_3}} \right)$$

And if it is dependent, all its solutions are in the form

$$\left(\mathbf{x}, \frac{\mathbf{s_1 - a_1 \cdot x}}{\mathbf{a_2}} \right) \quad \textbf{or} \quad \left(\mathbf{x}, \frac{\mathbf{s_2 - a_3 \cdot x}}{\mathbf{a_4}} \right)$$

where x is any real number whenever a_2 and a_4 are not zero. However, if a_2 and a_4 are zero, all its solutions are in the form

$$\left(\frac{\mathbf{s_1}}{\mathbf{a_1}}, y \right) \quad \textbf{or} \quad \left(\frac{\mathbf{s_2}}{\mathbf{a_3}}, y \right)$$

where y is any real number.

Using the above information, write a string function named `SystemAnalysis()` that takes six double parameters and two double reference parameters respectively. Given that first six (6) parameters correspond to a_1, a_2, a_3, a_4, s_1 and s_2 respectively, the function determines what type of system of linear equations the values of the parameters represent and does the following

- **if they represent an independent system, the function assigns the solution of the system to the reference parameters and returns the string `"independent"`.**
- **if they represent a dependent system, the function assigns the solution of the system for $x = 0$ to the reference parameters if the a_2 and a_4 parameters are not zero; otherwise, it assigns the solution of the system for $y = 0$ to the reference parameters. Then, it returns the string `"dependent"`.**
- **if the they represent an inconsistent system, the function assigns 0 to both reference parameters and returns the string `"inconsistent"`.**

Moreover, when the reference parameters are assigned values, the value of the first parameter should be the x value and the value of the second parameter should be the y value. For instance, the caller `SystemAnalysis(2,3,1,-6,8,-11,x,y)` will return `"independent"`, and `x` and `y` will be assigned `1` and `2` respectively.

Tracing

3. **In the commented section titled Tracing, construct a trace table (or list) of the caller `F(4,7)` where the definition of `F()` is below.**

```
int F(int a,int b)
{
  if(a <= 0 || b <= 0)
  {
    return 0;
  }
  int m = a + b, n;

  if(a < b)
  {
    n = a;
  }
  else
  {
    n = b;
  }
  m -= n;

  for(int i = m;i < m * n;i += m)
  {
    if(i % n == 0)
    {
      return i;
    }
  }
  return (n * m);
}
```

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: it returns the sum of the factors of the parameter if the parameter is positive or 0 if it is not positive*/`

```
01 | int O(string n)
02 | {
03 |   if(n <= 0)
04 |   {
05 |     return "0";
06 |   }
07 |   int S = n + 1;
08 |
09 |   for(int i = 2;i < n;i += 1)
10 |   {
11 |     if(n % i == 0)
12 |     {
13 |       s =+ i;
14 |     }
15 |   }
16 |   return s;
17 | }
```

b. `/*Intent: determines if exactly two of the parameters are equal to each other*/`

```
01 | bool K(int a,int b,int c)
02 | {
03 |   bool a = (a == b || a == c || b == c);
04 |   bool e = (a != b || a != c || b != c);
05 |   Return (d && e);
06 | {
```

c. `/*Intent: reads in a value and stores the input in the parameter until it is a digit*/`

```
01 | void G(const char& c)
02 | {
03 |   while(true)
04 |   {
05 |     cin >> c;
06 |
07 |     if(c >= '0' && c <= '9')
08 |     {
09 |       Break;
10 |     }
11 |
12 | }
```