Introduction to Computing
CS 151
Department of Physics and Computer Science
Medgar Evers College
Exam 4

Direction: Submit your typed work(s) as an upload(s) to the Exams directory of your GitHub repository or Dropbox, or in your Exam04 google classroom assignment.

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

Fundamentals

1. **For each of the following questions, write ONLY what is requested.**

   a. Define a bool function named `LCW()` that takes a string parameter. It returns true if the string parameter represents a lowercase word (i.e it only consists of lowercase letters); otherwise, it returns false.

   b. Define an int function named `MI()` that takes a double array parameter and an int parameter. Given that the int parameter represents the size of the array parameter, the function returns the index of the maximum value of the array parameter.

   c. Define a double function named `EPS()` that takes a double array parameter and an int parameter. Given that the int parameter represents the size of the array parameter, the function returns the sum of all the even positioned elements of the array whose values are positive.

   d. Define a void function named `CA()` that takes an int array parameter and an int parameter. Given that the int parameter represents the size of the array parameter, the function assigns each element of the array its position in the array.

   e. Define an int function named `RV()` that takes an int parameter. It should continually prompt the user to enter a number until the distance between the number and the parameter is at most 5; and then, it returns the input of the user.

Problem Solving

2. **Define a char function named `MaxOccurences()` that takes a string parameter. It returns the letter that appears the most in the parameter. However, if there are more than one letter that ties for the max appearances, the letter that appears later in the alphabet is returned. The case of the letter should be ignored. For instance, the function call `MaxOccurence("111MisSIssIpPi111")` will return 's'. Define any additional function you deem necessary.**

Tracing

3. Generate the trace table list for the function call `NC("46f81")` of the function below. Use a legend if needed.

```
string NC(const string wrd)
{
  string ky = "5814703692";
  string msg = "";

  for(int i = 0;wrd[i] != '\0';i += 1)
  {
    if(wrd[i] >= '0' && wrd[i] <= '9')
    {
      msg = msg + ky[wrd[i] - '0'];
    }
  }
  return msg;
}
```

Debugging

4. Write **ONLY** the line number and the entire line correction for each line that has an error in the code segment below.

```
01  struct Stack
02  {
03    char data[100];
04    int size;
05  }
06
07  bool IsFull(Stack* s)
08  {
09    Return (s->size == 100);
10  }
11
12  bool IsEmpty(stack* s)
13  {
14    return (s->size == 0);
15  }
16
17  char Top(Stack* s)
18  {
19    return data[s->size - 1];
20  }
21
22  void Push(Stack* s,char v)
23  {
24    if(!IsFull())
25    {
26      s->data[s->size] = v;
27      s->size += 1;
28    }
29
30
31  void Pop(Stack* s)
32  {
33    if(!IsEmpty(s))
34    {
35      s->size =- 1;
36    }
37  }
38
```

```
39  void Palin(string& word)
40  {
41    int i = 0;
42    Stack* v = New Stack;
43    v->size = 0;
44
45    for(word[i] != '\0';i += 1)
46    {
47      Push(v,word[i]);
48    }
49    i = 0;
50
51    while(!IsEmpty(v))
52    {
53      if(word[i] != Top(v))
54      {
55        word[i] = Top(v);
56      }
57      pop(v);
58      i += 1;
59    }
60    delete v;
61  }
```