Data Structures
CS 246
Department of Physics and Computer Science
Medgar Evers College
Exam 1

Direction: Modify the `"exam01.cpp"` file in your Exams directory of your GitHub repository; and then, submit your modified work in the Exams directory of your GitHub repository or Dropbox, or in your Exam01 google classroom assignment. You can only use the libraries included in the accompanying header file and the cpp file.

Problem	**Maximum Points**	**Points Earned**
1	5	
2	5	
3	5	
4	5	
Total	20	

Problems

1. Write the definition of the function `AverageDistance()` whose header is

```
double AverageDistance(Array<int>& data)
```

   It returns the average of the distances between adjacent elements of *data*. For instances, if *data* $= [1, 4, 6, 5, 9]$, it will return 2.5 since (3 + 2 + 1 + 4) / 4 equals 2.5. It is important to note that distance is never negative. Furthermore, it returns the element if *data* has a size of 1, and 0 if *data* is empty.

2. Write the definition of the function `Equal()` whose header is

```
template <typename T>
bool Equal(Array<T>& ar1,Array<T>& ar2)
```

   It returns true if the elements of *ar1* and *ar2* with the same index have the same value, and *ar1* and *ar2* have the same length; otherwise, it returns false.

3. Write the definition of the function `FrontAppend()` whose header is

```
template <typename T>
void FrontAppend(Array<T>& data,Array<T>& addon)
```

   It appends the content of *addon* to the beginning of *data*. It is important to note that you may have to resize *data* in order to hold both the original values from *data* and the values from *addon*. For instances, if *data* $= [a, b, c, d, e]$ and *addon* $= [f, g, h, i, j]$; then after the call of the function, *data* $= [f, g, h, i, j, a, b, c, d, e]$.

4. Construct the runtime table and calculate the worst-case scenario runtime for

```
void C(int a[],const int n,int k)
{
 int o[n];
 int c[10];

 for(int i = 0;i < 10;i += 1)
 {
  c[i] = 0;
 }

 for(int i = 0;i < n;i += 1)
 {
  c[(a[i] / k) % 10] += 1;
 }

 for(int i = 1;i < 10;i += 1)
 {
  c[i] += c[i-1];
 }

 for(int i = n - 1;i >= 0;i -= 1)
 {
  o[c[(a[i] / k) % 10] - 1] = a[i];
  c[(a[i] / k) %10] -= 1;
 }

 for(int i = 0;i < n;i += 1)
 {
  a[i] = o[i];
 }
}
```

where n is the size of a. Furthermore, assume the operation time cost is 1 for every operation. The table must be a comment.