



Data Structures
CS 246 - 040
Department of Physics and Computer Science
Medgar Evers College
Exam 1

Instructions:

- The exam requires completing a set of tasks within 45 minutes.
- Modify the accompanying cpp file. Write the nonprogramming tasks as comments in the file.
- Submit the modified cpp file to Github in the Exam01 directory and/or as an attachment on Google classroom under the Exam01 assessment.
- Lecture notes can be used.
- 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,
PRINT YOUR NAME AND THE DATE ON YOUR SUBMISSIONS

Grading:

Section	Maximum Points	Points Earned
Runtime	10	
Problem Solving	10	
Implementation	5	
Total	25	

Runtime

1. List the following big-O runtimes in ascending order.

$O(n)$	$O(\lceil \lg(n) \rceil)$	$O(1)$	$O(\lceil n \lg(n) \rceil)$	$O(\lceil \sqrt{n} \rceil)$
$O(\lceil \log(n) \rceil)$	$O(n^2)$	$O(2^n)$	$O(\lceil \lg^2(n) \rceil)$	$O(\lceil \lg(\lg(n)) \rceil)$

where $\lg(n) = \log_2(n)$, $\log(n) = \log_{10}(n)$ and $\lg^2(n) = (\log_2(n))^2$

2. Construct the runtime table and determine the runtime functions of the following function for the worst-case scenario. Let the cost of every operation be 1. Write the function in terms of n and state what n represents. You must use the ceiling or floor function for an accurate solution.

```
int F(const Array<int>& data)
{
    int c = 0;
    int x = data.Size() - 1;

    for(int i = 0; i <= x/2; i += 1)
    {
        if(data[i] == data[x-i])
        {
            c += data[i];
        }
    }
    return c;
}
```

Problem Solving

3. Write the definition of an int function named EvenOddDifference() whose header is

```
int EvenOddDifference(const Array<int>& data)
```

It returns the difference of the sum of the even values in *data* from the sum of the odd values in *data*. For instance, if *data* = [2,6,1,5,7,3], then the function will return 8.

4. Write the definition of a void function named InsertBefore() whose header is

```
template <typename T>
void InsertBefore(Array<T>& data, const T& target, const T& value)
```

If *target* is a value in *data*, the function will insert *value* into *data* immediately before the first instance of *target*; otherwise, it makes no changes to *data*. For instance, if *data* = ['h','f','j','f','q'], *target* = 'f' and *value* = 't', then after the call *data* = ['h','t','f','j','f']. Notice, if an insert is performed, the last element of *data* will be removed.

Implementation

5. Write a generic class named *Pair* that contains:

- public generic field named *key*.
- public int field named *value*.
- public default constructor that assigns the default value of the generic type to *key* and assigns 0 to *value*.
- public copy constructor.
- public assignment operator.
- public empty destructor.
- public string constant method named ToString() that takes no parameters. It returns a string in the format

<x,y>

where *x* and *y* are the values of *key* and *value* respectively.