Data Structures
CS 246 - 040
Department of Physics and Computer Science
Medgar Evers College
Exam 3

Instructions:

- **The exam requires completing a set of tasks within 60 minutes.**
- **Modify the accompanying cpp file. Write the nonprogramming tasks as comments in the file.**
- **The runtime table can be written in a spreadsheet.**
- **Submit all your work to Github in the Exam03 directory and/or as an attachment on Google classroom under the Exam03 assessment.**
- **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
Fundamental	5	
Runtime	5	
Tracing	5	
Problem Solving	5	
Implementation	5	
Total	25	

Fundamentals

Write ONLY what is requested.

a. What is a hash value?

b. What is a map data structure?

c. What is the meaning of a collision in hashing?

d. Which sorting algorithm performs the least amount of swaps for the worst-case scenario, and how many swaps will it perform if the array has a size of n?

e. How does the open-addressing insertion method deal with collisions?

Runtime

Construct the runtime table that includes a statement column 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, which is the size of the array. You may need to use the ceiling or floor function for an accurate solution.

```
void D(Array<string>& data)
{
  for(int i = 0;i < data.Size();i += 1)
  {
    data[i] = "[";

    for(char j = '0';j != '9';j += 1)
    {
      data[i] += j;
    }
    data[i] = "]";
  }
}
```

Tracing

Write an array trace table for any two of the three sorting algorithms discussed in class [bubble sort, insertion sort, selection sort] that provides only the swaps that will be performed on the array data $= [\mathbf{2}, \mathbf{3}, \mathbf{7}, \mathbf{4}, \mathbf{8}, \mathbf{1}]$. Each trace table must start with the initial value of data.

Problem Solving

Write the void function `InsertionSort()` whose header is

```
template <typename T>
void InsertionSort(Node<T>* root)
```

Its definition is the insertion sort algorithm implemented with a linked list. Remember a linked list can be empty.

Implementation

Given that the fields of a class named *HashMap* is

```
template <typename V>
class HashMap
{
  Node<Pair<int,V>>* slots[200];
};
```

write the following methods

- **private int method `hash()` that takes an int parameter. It should implement the division method algorithm for hash functions using the absolute value of the parameter.**
- **public bool method named `Contains()` that takes an int parameter named *key*. It returns true if *key* is in the hashmap; otherwise, it returns false.**