TIC2001 Data Structure and Algorithm Quiz 1
Name: ______________ Student#:_A____________

Question 1 (5 x 3 marks)

What is the time complexity of calling the functions on the right column in terms of n for n > 0? Give your answer in ***the Big O notation***. The function `doSomething(m)` will have a time complexity of O(m) depending on the input m.

`void f(int n) {`
`  for (int i=0;i<=n;i++)`
`    doSomething(i);`
`}`
	Time complexity of **`f(n)`** =
	$O(n^2)$

`int f(int n) {`
`  for (int i=0;i<=log(n);i++)`
`    doSomething(1);`
`  for (int j=0;j<=n*n;j++)`
`    doSomething(1);`
`}`
	Time complexity of **`f(n)`** =
	$O(n^2)$

`void f(double n) {`
`  for(int i=n;i>=n/2;i--)`
`    for(int j=0;j<n;j++)`
`      doSomething(n);`
`}`
	Time complexity of **`f(n)`** =
	$O(n^3)$

Question 2 Sorting Detective (5 x 3 marks)

Your goal is to identify which sorting algorithm is being used in which case. The first row contains an unsorted list of numbers. The last row contains a sorted list of numbers. Each intermediate rows contains a partially sorted list of numbers.

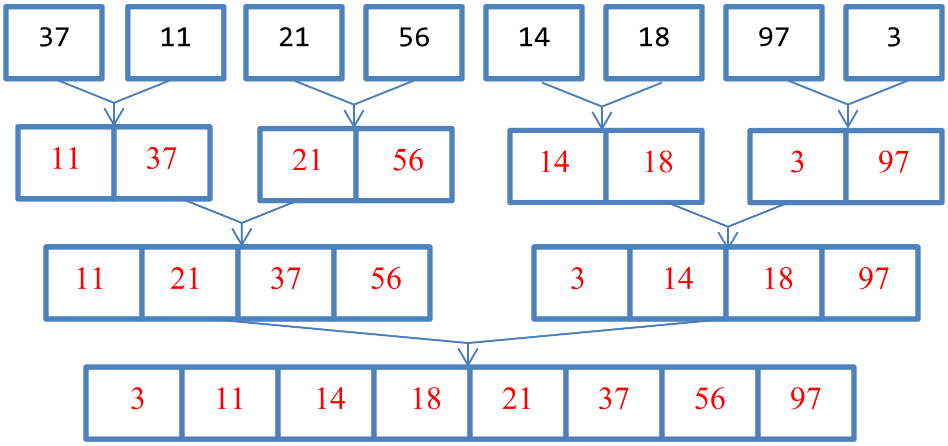
Each intermediate row was constructed by beginning with the unsorted list and running one of the sorting algorithms that we learned about in class, stopping at some point before it finishes. Identify, below, which row was (partially) sorted with *which* algorithm.

Hint: Do not just execute each sorting algorithm, step-by-step, until it matches one of the columns. Instead, think about the invariants that are true at every step of the sorting algorithm. Name the algorithms on the right column for each row. Your choices are: Bubble/Insertion/Selection/Quick/Merge Sorts.

Unsorted	6	3	1	4	8	7	5	2	**Algorithm**
A	3	1	4	2	5	6	8	7	QuickSort
B	3	1	4	6	5	2	7	8	BubbleSort
C	1	3	4	6	2	5	7	8	MergeSort
D	1	2	3	4	8	7	5	6	SelectionSort
E	1	3	6	4	8	7	5	2	InsertionSort
Sorted	1	2	3	4	5	6	7	8	

Question 3 MergeSort(15 marks)

Perform a merge sort for the following array according to ***ascending*** order. Show the steps.



Questions 4 True and False Questions (5 x 3 marks)

	Answers:
A private member/attribute of a class X in C++ is impossible to be accessed by other classes. (A private member/attribute can be accessed by other classes with "friends".)	F
If we declared "`int **ptr;`", `ptr` is a pointer of pointer to integer.	T
If we declared "`int arr[10];`", `arr` is a pointer to integer.	T
MergeSort will be slower if the data is stored in linked lists than stored in arrays.	F
We cannot do an "operator overload" on the arithmetic operations (`+`, `-`, `*`, `/`) in C++.	F