B. Math III Year
2005 Mid-term Examination – CS III (Max. marks: 50 and Time 3 hours)

Q1 [20 Marks: 2 Marks for each part] Give ***very brief*** answers
a. State true or false: "Merge sort runs faster if the input is already sorted"
b. What is the worst case runtime for the Quick sort algorithm?
c. What are the best case, worst case, and average case order for the radix sort algorithm?
d. What is the order of inserts and deletes on arrays (assume that array has n elements)?
e. Arrange the following in the correct increasing (fastest to the slowest algorithm) order: $O(n), O(2^n), O(n^3), O(n \log n), O(n^2), O(\log n), O(n!)$
f. Which data structure would be most appropriate for representing path of a traveling salesperson who will take the reverse path on the return trip?
g. What is the order of the Quick Sort algorithm if the largest value in the list is always chosen as the pivot?
h. Data types are important for a number of reasons. We discussed four reasons in the class. State these four reasons (use one word to describe each reason).
i. What is the main reason for recursive algorithms for functions such as Fibonacci numbers to be slow?
j. The decision to use recursive vs non-recursive methods usually lead to which trade-off?

Q2 [5 Marks] This question refers to the code in the table below:

There are some errors (logical as well as syntax) in this code. Identify and correct them (refer to the line numbers in your answer).
What is this program attempting to do?

```
1.  #include <stdio.h>
2.  main()
3.  {
4.     char S[80];
5.     printf("Enter
       text:");
6.     gets(S);
7.     do_something(S)
8.  }
```

```
9.  do_something(S)
10. {
11.   if (S != NULL)
12.    {
13.       do_something(S++);
14.       putchar(*(--S));
15.    }
16. }
```

Q3 [10 Marks] The complexity for many divide and conquer algorithms is given by a recurrence relation:

$T(n) = T(1)$ for $n=1$ and $T(n) = aT(n/b) + f(n)$ for $n > 1$

where a and b are known constants, T(1) is known, and n is a power of b. Obtain a closed form expression for T(n) as a function of n, f(n) and T(1).

Q4 [5 Marks] Write the algorithm (follow the pseudo-code conventions discussed in the class) which uses stacks to evaluate a postfix expression.

Q5 [10 Marks] Write an algorithm for a function that will accept a sorted integer array and its length as inputs (the array may have duplicate elements). The function should compact it and then return the new length of the array. Also state the order of your algorithm. For example if the input array contains: 30, 34, 39, 47, 47, 59, 69, 69, 70 when the function returns, the contents of the array should be: 30, 34, 39, 47, 59, 69, 70 with a length of 7 returned.