DATA ANALYSIS MIDTERM EXAM (November 28, 2012)

Q1) Write a C function that

a) Finds the maximum and minimum values in the linked list,

b) Deletes the first and penultimate nodes from the list.

(Nodes have integer values ant pointer of the first node is known)

Q2)

a) Write a C function that finds the sum of the values of non-leaf nodes in the binary search tree,

b) Write a C function that deletes the node with the smallest value.

(Nodes have integer value and pointer of the root is known.)

Q3)

a) Using respectively the given velues, construct a AVL tree and delete the root of the tree.

10, 85, 15, 70, 20, 60, 30, 50, 65, 80, 90

b) Given the mathematical expression

(A + B) ^ D / E / F * G – Z * (X – Y).

Find the postfix expression by using stack and construct binary tree for the expression.

Q4) Explaining the C program given back page, find the printed values.

```
#include <stdio.h>
#include <conio.h>
typedef struct list{
   int value;
  struct list *next;
  }LST;
int main()
 {
   LST *start;
 int A[6] = {4, 7, 1, 3, 6 ,2}, i;
 void insert1 (int, LST*);
 LST* insert2 (int, LST*);
 void print( LST* );

 start = (LST *)malloc( sizeof (
LST ) );
 printf("\n Insert last digit of
your school number...");
 scanf("%d", &( start -> value ));
 start -> next = NULL;

 for( i = 0; i < 6; i++)
  {
   if( start -> value > A[i] )
     {
       insert1( A[i], start);
       print( start );
     }
    else
      {
       start = insert2( A[i], start);
        print( start );
       }
  } //end of for
}//end of main
```

```
/*******************/
void insert1( int x, LST*ptr)
 {
  while(ptr -> next != NULL)
    ptr = ptr -> next;

    ptr = ptr -> next =
(LST*)malloc(sizeof(LST));
    ptr -> value = x;
    ptr -> next = NULL;
 }
/*******************/
LST* insert2(int x, LST* start)
 {
   LST*ptr;
   ptr = ( LST* )malloc(
sizeof(LST) );
   ptr -> value = x;
   ptr -> next = start;
   start = ptr;
   return( start );
  }
/*******************/
void print( LST *ptr )
{
  printf("\n\n");
  while( ptr != NULL)
   {
    printf(" %d ",ptr -> value );
    ptr = ptr -> next;
   }
}
```