Data Structures and Algorithms

Mid-Sem Examination is to be conducted in Weebox.

Total marks 20. Weightage 20%.

All questions are compulsory.

The question paper is divided into **two sections**:

Section A - contains 8 MCQ type questions each of 1 mark.

Section B - contains 6 questions each of two marks. You must solve these questions with pen and paper, if required, and write the answer in the provided text box.

The written answer will be **auto-evaluated** with a string matching algorithm. You must be very cautious about the answers you are writing. Follow the instructions carefully.

Example Question - Section A.

Which of the following algorithms is best designed by recursion?
A. Tower of Hanoi
B. Bubble sort
C. Linear search
D. Insertion sort

Answer: A

Example Questions - Section B.

Be careful about what you write. Look at the case of the letters and don't give any space in between. No leading or trailing spaces are allowed.

Variables are to be specified in the EXACT way already mentioned in the pseudo-code.

Q1. Consider the following infix expression: **(a+b)*(c-d)**.

Write the corresponding postfix expression in the box provided.

Correct Answer

ab+cd-*

Note that if you write the answer in the following way it will be ***incorrect.***

Ab+cd-*

The answer is case-sensitive

Or

a b + c d - *

No leading or trailing space, no space in between the letters is allowed.

Q2. Consider the following pseudo-code for inserting a node at a Singly-linked list before a specific node (the node that contains the element **before** in its data). Fill the boxes with the missing statements.

insert_beore (head, before, item)

```
  {
    currNode = head;
    prevNode = head;
    new_node = (NODE*)malloc(sizeof(NODE));
    new_node ->data=item;

    while (              )
    {
      prevNode = currNode;
      currNode = currNode->next;
    }
     prevNode->next = new_node;



  }
```

Answer:

You need to fill the first box as follows:

currNode->data!=before

This is a condition, there is no space in between the letters.

You need to fill the second box as follows:

new_node->next=currNode;

This is a statement ending with a semi-colon. As all other statements in the algo are also ending with a semi-colon, you need to put the semicolon at the end.

Q3. Consider the following pseudo-code for inserting a node at the last of a Singly-linked. Fill the box with the missing statement.

```
insert_last(head, item)
 {
   currNode = head
   new_node = (NODE*)malloc(sizeof(NODE))
   new_node ->data=item
   new_node ->next = NULL

   while (                )
     {
       currNode = currNode->next
     }


 }
```

You need to fill the first box as follows:

currNode->next!=NULL

Notice we mentioned NULL in capital letters and not as null or Null. Notice now we have not put any semi-colon at the end of the statement. This is because, this entire pseudo-code does not contain any semi-colon at the end of any statement.

You are not allowed to mention currnode or CurrNode or Currnode. Look at the name of the variable, they have to be twitten in exactly the same way mentioned in the algorithm – correct way – currNode

Also there will be no space in between any letter. You are not allowed to mention the statement as currNode -> next != NUL (space in between is not allowed.)

You need to fill the second box as follows:

currNode=new_node

Notice now we have not put any semi-colon at the end of the statement. This is because, this entire pseudo-code does not contain any semi-colon at the end of any statement.