



United International University (UIU)
Dept. of Computer Science & Engineering (CSE)
Midterm Year: 2017 Trimester: Summer
Course: CSI 217 Data Structures, Marks: 30, Time: 2 hours

There are SIX questions. Answer any FOUR. Figures in the right-hand margin indicate full marks.

-
1.
 - a) Define data Structure. 1.0
 - b) Can you write a sorting algorithm which will perform better if most of the elements are sorted? Explain your answer too. 2.0
3.5
 - c) Show the working mechanism of descending order Insertion Sort for the data given below
17 15 16 19
 2.
 - a) For searching a word in English to Bengali Dictionary which search is better and why? 2.0
 - b) Design a Recursive Linear Search algorithm. 2.0
 - c) Show the mechanism of Recursive Binary Search algorithm for the search key 80 using the descending order data mentioned below. 3.5
100 90 80 70 65 60 50 40 30 20 10
 3.
 - a) Find the position of the first partitioning element using the following instance of the descending order Quick Sort algorithm. 3.5
27 18 9 85 50 16 32 15 87 76 30 17
 - b) Explain the working mechanism of ascending order Selection Sort with an example. 2.0
 - c) Distinguish between Replacement and Bubble Sorting algorithms. 2.0
 4.
 - a) What are the merits of dynamic memory allocation over the static allocation? 1.0
 - b) Discuss the merits and demerits of a linked list data structure over an array. 2.0
 - c) Declare a variable for the linked list data structure in programming language C. 2.0
Student(name, id, marks, , flink)
Where, **name** is a **string** field
id is an **integer** field
marks is a **float** field
flink field contains the address of the next node in the linked list
 - d) Show the diagrams of different types of linked lists. 2.5
 5.
 - a) Design a code segment to display the sum of even data stored in a linear linked list. 4.5
 - b) Write a code segment to concatenate two linear linked lists. 3.0

6. Draw a diagram for each of the statements given below using the following structure

7.5

```
struct list{
    int data;
    struct list *next;
};
typedef struct list node;
node *tempprev, *temp, *tempsuc, *temp1;
```

Statements:

```
tempsuc=(node*)malloc(sizeof(node));
temp1=(node*)malloc(sizeof(node));
tempprev=(node*)malloc(sizeof(node));
temp=(node*)malloc(sizeof(node));
temp->next=tempprev;
temp->next->next=tempsuc;
tempsuc->next=temp;
tempsuc->next->next=temp1;
tempsuc->next->next->next=temp->next;
temp->data=15;
temp1->data=20;
tempprev->data=35;
tempsuc->data=17;
```