United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

Midterm Year: 2017 Trimester: Spring

Course: CSI 217 Data Structures, Marks: 30, Time: 2 hours

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

1.	a)	Define data Structure. Discuss the importance of data structure in computer applications.	2.0
	b)	For searching a string from an array of strings which searching method will perform better and why?	2.0
	c)	Suggest a data structure for Binary Search. Show the simulation of Binary Search algorithm to find 80 and 85 using your data structure for the following data: 32, 39, 45, 48, 56, 68, 71, 72, 78, 81, 83, 85, 90, 95	4.0
	d)	Explain the working mechanism of Selection Sort with an example.	2.0
2.	a)	When Bubble Sort algorithm can perform better than Quick Sort algorithm?	1.0
	b)	Write an algorithm for Quick Sort to sort the data in descending order.	3.0
	c)	Execute ascending order Quick Sort Algorithm up to second partitioning elements. 8, 18, 11, 9, 22, -5, -9, -11, 28, 11	3.0
	d)	Design a recursive algorithm for Insertion Sort.	2.0
	e)	What are the differences between replacement and bubble sort?	
3.	a)	Declare a variable for the linked list data structure in programming language C ***Student(name, id, marks, next)*** Where, **name** is a **string** field **id** is an **integer** field **marks** is a **float** field **next** field contains the address of the next node in the linked list Write an algorithm to insert an element in any place of the list.	4.0
	b)	Design a code segment to implement the linear search algorithm using linked list.	3.0
	c)	Implement a linear linked list of integer elements and add the elements of the list.	3.0

4. a) What are the advantages of linked list over array? 1.0

b) Draw a diagram for each of the following statements using the following structure 6.0

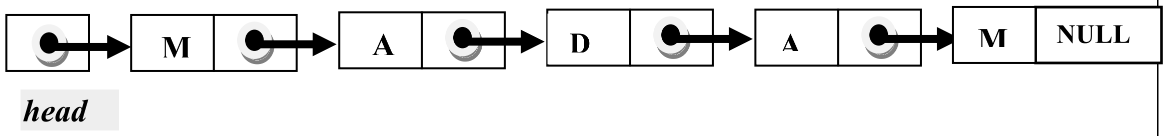
```
                    struct list{
                              int data;
                              struct list *next;
                             };
                    typedef struct list node;
                    node *tempprev, *temp, *tempsuc;
```

Statements:

```
tempprev=(node*)malloc(sizeof(node));
temp=(node*)malloc(sizeof(node));
tempsuc=(node*)malloc(sizeof(node));
tempprev->data=5;
tempsuc->data=17;
temp->data=15;
temp->next=tempprev;
temp->next->next=tempsuc;
tempsuc->next=temp;
free(tempsuc);
temp->next->next=temp;
temp1=(node*)malloc(sizeof(node));
temp1->data=20;
temp->next=temp1;
temp1->next=tempprev;
```

c) Suppose, we have a linear linked list named ***head*** where each node contains an integer value and a pointer to the next node (see the example below). Write an algorithm to check whether this list contains palindromic character sequence or not. For a palindromic character sequence, we get the same sequence if we reverse the original sequence. 3.0

Example:


M
A
D
A
M
NULL
head