# United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

Exam: Midterm Year: 2018 Trimester: Fall

Course: CSI 217 Title: Data Structure

Marks: 30 Time: 1 hour 45 minutes

## Answer all the following questions.

**1. a)** Compare advantages and disadvantages of linked list over Array with proper example. **2.0**

**b)** Suppose that you are playing a game called "guess the number" with your friend and you have to guess a number within a range of 1 to 100 (inclusive). To have a hint, you can follow two methods: **3.5**

i. Ask your friend "is the number 1"? And if s/he says no, then ask "is the number 2", and go on.

ii. Ask your friend whether the number is greater or smaller than 50? If s/he says greater than 50. Then you ask whether the number is greater or smaller than 75? And go on…

Which of the above method is more efficient if your friend guessed the number 100? With how many questions you can answer that s/he guessed 100 in both (i) and (ii)? Write a pseudo-code for guessing the number with the most efficient way.

**c)** Design a recursive algorithm for linear search. **2.0**

**2. a)** Declare a variable for linked list data structure in C programming language. **2.0**

***Student (name, id, marks, birthday (day, month, year), next)***

Where, name is a **string** filed, id is an **integer**, marks is **float** type, birthday is another data structure which contains day, month and year field of **integer** type and **next** filed hold the address of the next node of the list.

**b)** Construct a code that counts the total number of nodes in a linked list. **2.0**

**c)** Write a function that removes every alternate element of a singly linked list from the beginning. The before and after states of the linked list will be as follows. **3.5**

| Before | After |
|---|---|
| 1->2->3->4->5->6->7->8->NULL | 1->3->5->7->NULL |
| 1->2->3->NULL | 1->3->NULL |
| 1->NULL | 1->NULL |

Assume that, the node is a structure that looks like as follows. Also assume that, there is data already inserted in the linked list and head is the pointer to the start of the linked list. You just have to write a function called **removeAlternateElements()** to do the task.

```
struct node{
    int data;
    struct node *next;
} ;
struct node* head;
void removeAlternateElements(){

}
```

3. a) Apply quick sort algorithm for the following dataset up to second partition. **2.5**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 27 | 7 | 12 | 5 | 100 | 1 | 2 | 8 | 10 | 15 |

b) Write an algorithm for Selection Sort. **2.0**

Or

Apply Insertion sort for the following data. Show each steps.

| | | | | | | |
|---|---|---|---|---|---|---|
| 12 | 12.2 | 11.5 | 11.25 | 10 | 9.5 | 100 |

c) Consider you have the following linked list. **3.0**

12->20->10->14->20->NULL

Design an algorithm that delete the node that contain highest value. Assume that the linked list is not sorted.

4. a) Implement Push and Pop functions of STACK using Array and Linked list. **3.0**

b) Draw a diagram for each of the statements given below. You need to draw a picture for each of the instructions that has a line number at their left. **4.5**

```
1.  struct list
2.  {
3.     int data ;
4.     struct list *next ;
5.  } ;
6.  list *tempprev, *temp, *tempsuc ;
7.    tempprev = (list*) malloc(sizeof(list)) ;
8.    temp = (list*) malloc(sizeof(list)) ;
9.    tempsuc = (list*) malloc(sizeof(list)) ;
10.   tempprev->data = 5 ;
11.   tempsuc->data = 10 ;
12.   temp->data = 100 ;
13.   temp->next = tempprev;
14.   temp->next->next = tempsuc;
15.   tempsuc->next=temp;
16.   free(tempsuc);
17.   temp->next->next = temp ;
18.   list *temp1 = (list*) malloc(sizeof(list)) ;
19.   temp1->data = 200 ;
20.   temp1->next = temp;
```