United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

Assignment Year: 2021 Trimester: Spring Program: BSCSE

Course: CSE 2215/CSI 217 Data Structure and Algorithms-I

-

There are EIGHT questions. Answer all of them. Please submit assignment by 12/05/2021 Wednesday within 11:55pm(night) to lms.

1. Show the mechanism of Descending Order Insertion Sort Algorithm on the following data. Here DD means last two digits of your student ID. [2]

   8 20 9 DD

2. Show the mechanism of Descending Order Selection Sort Algorithm on the following data. Here DD means last two digits of your student ID. [2]

   8 20 9 DD

3. Show the mechanism of Descending Order Recursive Binary Search Algorithm on the following data where search key= LAST_TWO_DIGIT_OF_YOUR_STUDENT_ID [3]

   80 70 60 50 40 30 20 10

4. Implement logic to create a doubly linked list using all the individual digits of your own STUDENT ID. Use the following structure and variables to design the logic. [3]

```
struct list{
          int data;
          struct list *next;
          struct list *back;
        };
typedef struct list node;
node *start, *temp;
```

   For example, STUDENT ID=011202074
   Individual digits are 0, 1, 2, 4, 7
   Linked List is: **NULL←0↔1↔2↔4↔7→NULL**
   Assume starting node is “start”

5. Show the effects of each of the following statements. Assume, x=LAST_DIGIT_OF_YOUR_STUDENT_ID. Here, the following structure and variables are used. [3]

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

```
super=(node*)malloc(sizeof(node));
super->next=super;
super->data=x*3;

temp=(node*)malloc(sizeof(node));
temp->data=super->data+ x*5;
temp->next=super;

temp1=(node*)malloc(sizeof(node));
temp1->data=super->data * temp->data;
temp1->next=super;

super->next=temp1;
super->next->next=temp;
```

6. Find time complexity of the following iterative algorithm. [2]

```
for i=1 to n do
  sum=0
  for j=1 to n do
     sum=sum+i+j
  end for
end for
```

7. Find time complexity of the following recursive algorithm. [2]

```
int factorial (int n){
    if (n==1)
        return 1;
    else
        return n*factorial(n-1);

}
```

8. Write an algorithm to reverse an input string using a STACK. [3]