Question 1 **(25 marks)**

a) What is the best data structure that can be used to reverse a given string? State reasons.
(3 marks)

b) Constructor of the stack class is implemented as follows,

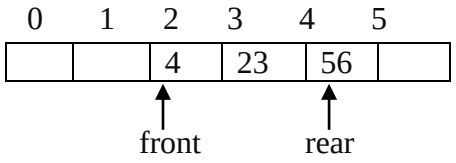
```
public StackX()
{
   stArr = new double[10];
   maxSize = 10;
   top = -1;
}
```

i. State a disadvantage of using the above constructor. (3 marks)

ii. Rewrite the constructor to avoid the disadvantage stated above. (2 marks)

c) Implement the **`size()`** method to return the number of elements in the stack.
(3 marks)

d) Consider the following **circular queue** frame and draw the resultant queue frame after executing following code segment.
(3 marks)




```
q1.insert(66);
q1.insert(56);
q1.insert(q1.remove());
```

e) Consider the given **`remove()`** method implemented for a circular queue. This code contains some errors. Write the line numbers with errors and correct them.

(6 marks)

```
1. public int remove()
2. {
3.      if (rear == -1)
4.      {
5.           System.out.println("Queue is empty");
6.           return false;
7.      }
8.      else
9.      {
10.          int temp = queueArray[front++];
11.          nItems--;
12.          return temp;
13.     }
14.}
```

f) "rear == -1condition can be used to find whether a linear queue is empty" . Do you agree with this statement? Justify your answer.

(5 marks)

Question 2 **(25 marks)**

a) Consider the following LinkedLists and write a code segment that would change the LinkedList from diagram A to diagram B. (First, P1, and P2 are references to given links)

(6 marks)

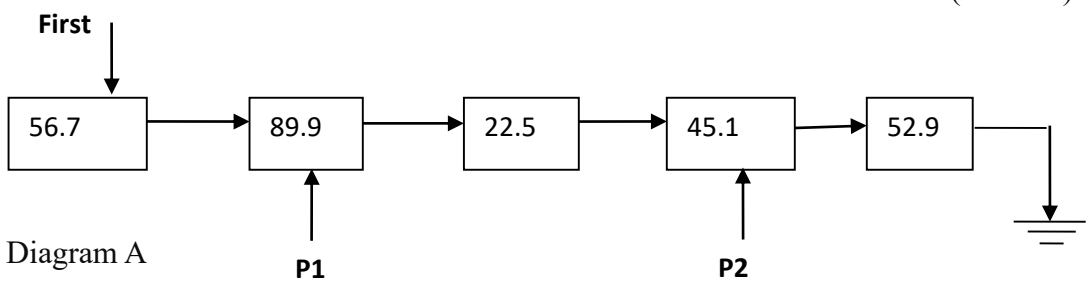


Diagram A

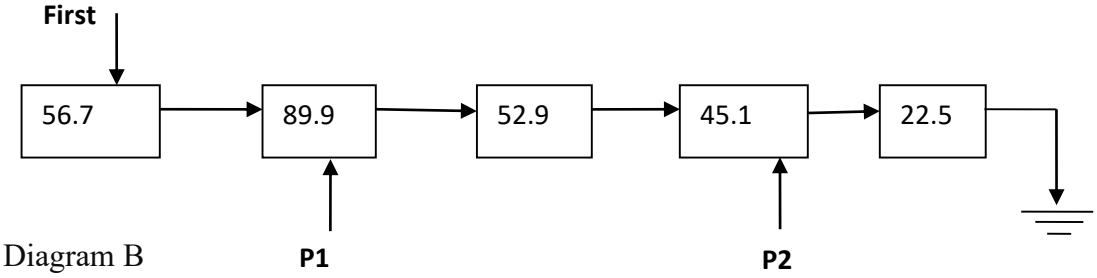


Diagram B

b) A company maintains the items of their store using a linked list. An item consists of

Item Code	integer
Unit Price	double

Assume the **item** class and the **linked list** class has already been implemented.

i) Implement a method called `findUnitPrice()` in the linked list class to return the unit price when the item code is given as a parameter.

(5 marks)

ii) State the modifications to be done to make the above linked list a doubly linked list?

(2 marks)

c) i) Insert the following data in to a Binary Search Tree. (4 marks)

m g k s v i d t

ii) Complete the below method called displayDesc() in the tree class to display the data in descending order. (3 marks)

```
private void displayDesc(Node localRoot)
{
        if (localRoot != null)
        {
                ……
                ……
                ……
        }
}
```

iii) Implement a method to return the maximum value from a binary search tree (5 marks)

Question 3 **(25 marks)**

a) Consider the linear search algorithm given below

```
LINEAR-SEARCH(A, v)
1. for i = 1 to A.length
2.       if A[i] == v
3.           return i
4.   return NIL
```

i) Identify the worst case and best case scenario in linear search algorithm (4 marks)

ii) Find the worst case and best case running time in Big O notation. (2 marks)

b) Consider the merge sort algorithm given below

MERGESORT (A, p, r)

1. **if** p < r
2. q = $\lfloor$(p+r)/2$\rfloor$
3. **MERGESORT** (A, p, q)
4. **MERGESORT** (A, q+1, r)
5. **MERGE** (A, p, q, r)

MERGE(*A*, *p*, *q*, *r*)

1 $n_1 = q - p + 1$
2 $n_2 = r - q$
3 create arrays $L[1..\ n_1 + 1]$ and $R[1..\ n_2 + 1]$
4 **for** $i = 1$ **to** n_1
5 $L[i] = A[p + i - 1]$
6 **for** $j = 1$ **to** n_2
7 $R[j] = A[q + j]$
8 $L[n_1 + 1] = \infty$
9 $R[n_2 + 1] = \infty$
10 $i = 1$
11 $j = 1$
12 **for** $k = p$ **to** r
13 **if** $L[i] \leq R[j]$
14 $A[k] = L[i]$
15 $i = i + 1$
16 **else** $A[k] = R[j]$
17 $j = j + 1$

i) In the MERGE algorithm, what is the purpose of line 8 and 9. Explain the reason for having ∞ (6 marks)

ii) Find the recurrence equation for mergesort algorithm and explain the way you derived it (3 marks)

iii) What is the running time of merge sort? (2 marks)

c) One of the main tasks of the operating system (OS) is to schedule processes for Input Output (IO) devices. When there are several request for IO devices from several processes, OS can create a queue and insert processes into the queue. Assume the queue implemented by the OS is a **priority queue** and numbers are assigned to each process to represent priority with the high number means high priority. Currently there are 8 processes waiting for the IO devices in the priority queue with the priority given below at time t_0.

Process Number	1	2	3	4	5	6	7	8
Priority	150	120	130	24	30	9	4	2

i) Briefly explain which process will get the IO device first? (1 mark)

ii) Which process will get the IO device next? (1 mark)

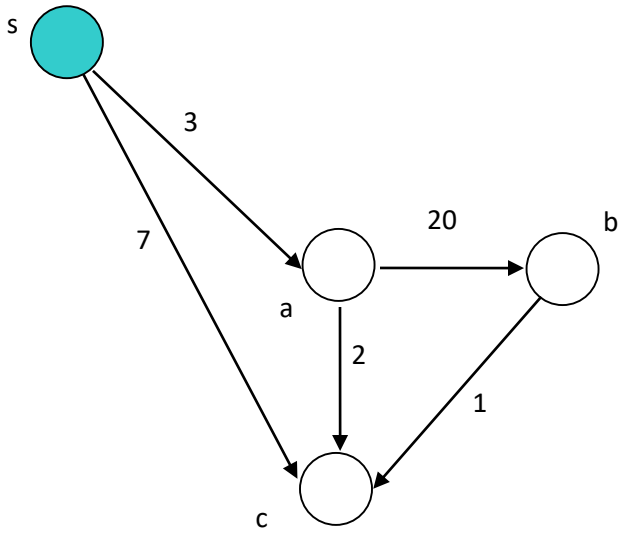
iii) If the new process has come to the queue at the time t_0 with the priority number of 200, Briefly explain how does the OS will insert this process into the queue with the help of Max_**Heap_Insert()** algorithm given.

(6 marks)

```
Procedure  MAX_HEAP_INSERT(A, key)
1. A.heap_size = A.heap_size +1
2. i = A.heap_size
3. while i > 1 and A[PARENT(i)] < key
4.       A[i] = A[PARENT(i)]
5.        i = PARENT(i)
6. A[i] = key
```

Question 4 **(25 marks)**

a) Apply the **Dijkstra's** algorithm given to find the shortest path from the source vertex **s** to all the other vertices of the graph. (For the purpose of illustration, assign the values only once to the given algorithm and use diagrammatic way to reach the answer.)
(6 marks)



INITIALIZE-SINGLE-SOURCE(*G*, *s*)
1 **for** each vertex $v \in G.V$
2 $v.d = \infty$
3 $v.\pi$ = NIL
4 $s.d = 0$

RELAX(*u*, *v*, *w*)
1 **if** $v.d > u.d + w(u, v)$
2 $v.d = u.d + w(u, v)$
3 $v.\pi = u$

DIJKSTRA(*G*, *w*, *s*)
1 INITIALIZE-SINGLE-SOURCE(*G*, *s*)
2 $S = \emptyset$
3 $Q = G.V$
4 while $Q \neq \emptyset$
5 u = EXTRACT-MIN(Q)
6 $S = S \in \{u\}$
7 for each vertex $v \in G.Adj[u]$
8 RELAX(*u*, *v*, *w*)

b)

i) If modulo value is q = 100, how many spurious hits and valid hits do the **Rabin - Karp matcher** encounter in the text T = 10300100200100 when looking for pattern P = 100? (4 marks)

ii) How do you reduce the number of spurious hits in (1)? (2 marks)

iii) What should be the number of spurious hits and valid hits if the best-case scenario occurs in Rabin-Karp algorithm?

(2 marks)

c) Draw the state transition diagram for a string-matching automation for the pattern P = ***abb*** and take the input alphabet as {*a*,*b*}

(6 marks)

d) Following is the **Naïve-String-Matcher** algorithm, which is used to find the occurrence(s) of a pattern string within another string or body of text.

```
Naïve-String-Matcher (T, P)
1.    n = T.length
2.    m = P.length
3.        for s = 0 to n-m
4.            if P[1..m] = T[s+1..s+m]
5.                print "Pattern occurs with shift" s
```

Given the text and pattern as follows;

Text ***T***

a	b	a	b	a	a	a	a	b

Pattern ***P***

a	b	a

i) How many comparisons would occur in this algorithm? (2 marks)

ii) Show that the worst-case time complexity of the above algorithm is $O(m(n-m+1))$ where n is the number of characters in the text and m is the number of characters in the pattern. (3 marks)

End of the Paper