Exam Date: 23/9/2022 | MSE Solution | ACAD. YR: 2022-23.

SEM 3. | DATA STRUCTURE | (20CP201T).

---

**Q1.** Multiple Choice Questions.

A) ii

B) iii

C) O(1) i

D) i – effective use of memory

E) ii) REAR = (REAR+1) % CAPACITY.

2 Marks for each correct answer.

0 Marks otherwise

---

**Q2 (A)** Differentiate linear and non-linear D.S.

[Any 5 points for 3 Marks each].

| Sl. No. | Linear D.S. | Non-Linear D.S. |
|---|---|---|
| 1. | Elements are arranged in linear order. | 1. Elements are arranged in hierarchical manner. |
| 2. | Comparatively easier implementation | 2. Implementation complex as compared to linear |
| 3. | Elements can be traversed in a single run only. | 3. Elements can not be traversed in single run. |
| 4. | Memory is not utilized in an effiuent way. | 4. Memory is utilized in an efficient way. |
| 5. | Examples :- Array, Stack, Queue, Linked List | 5. Examples :- Trees, Graphs |
| 6. | Applications :- Software Development | 6. Applications :- AI, ML, Image Processing. |

Q1 B) Algo / Pseudocode to evaluate postfix expression

This algorithm finds the VALUE of an expression 'P' written in postfix notations.

(1) Add a parenthesis ")" at the end of P. [1 Marks]
[This acts as sentinel]

(2) Scan 'P' from left to right and [1 Marks]
Repeat steps (3) and (4) for each element of 'P'
until sentinel ")" is encountered.

(3) If an operand is encountered, then; [1 Marks]
push it on STACK.

(4) If an operator ⊗ is encountered, then;
(a) Remove the top two elements of STACK, where
A = top element & B = next to top element
(b) Evaluate B⊗A. [1 Marks]
(c) Push the result of (b) back on STACK.

[End of If structure]

[End of step 2 loop]

(5) Set VALUE equal to the top element on STACK. [1 Marks].

(6) Exit.

[5 Marks for 5 steps].

Q3 (A)

## Priority Queue.

i) Definition and Concept } ⟹ 2 Marks.

ii) Priority Queue with example } ⟹ 3 Marks.

---

Q3 (B). Given expression - A + B * (C - D) / E + F - G.

| Symbol Scanned | STACK. | Post fix Notation. | |
|---|---|---|---|
| A | ( | A | |
| + | (+ | A | |
| B | (+ | AB | |
| * | (+* | AB | |
| ( | (+*( | AB | |
| C | (+*( | ABC | |
| - | (+*(- | ABC | |
| D | (+*(- | ABCD | |
| ) | (+* | ABCD- | |
| / | (+/ | ABCD-* ⟹ Imp. | 1 Marks. |
| E | (+/ | ABCD-*E | |
| + | (+ | ABCD-*E/+ ⟹ Imp | 1 Marks. |
| F | (+ | ABCD-*E/+F | |
| - | (- | ABCD-*E/+F+ ⟹ Imp | 1 Marks. |
| G | (- | ABCD-*E/+F+G | |
| ) | Empty | ABCD-*E/+F+G- ⟹ Final Answer. | 2 Marks. |

5 Marks for complete correct answer.

Otherwise partial Marks based on percentage of correctness.

Q4

| Operations | Output | Stack | Queue |
|---|---|---|---|
| push (A) | - | A | - |
| push (B) | - | AB | - |
| enq (C) | - | AB | C |
| pop () | B | A | C |
| enq (D) | B | A | CD |
| push (E) | B | AE | CD |
| deq () | BC | AE | ·D |
| push (F) | BC | AEF | ·D |
| enq (G) | BC | AEF | ·DG |
| deq () | BCD | AEF | ··G |
| push (H) | BCD | AEFH | ··G |
| deq () | BCDG | AEFH | ··· |
| push (I) | BCDG | AEFHI | ··· |
| deq () @ 11 | BCDG | AEFHI | ··· |
| deq () empty | BCDG | AEFHI | ··· |
| push (J) stack full | BCDG | AEFHI | ··· |
| pop () | BCDGI | AEFH | ··· |
| enq (K) | BCDGI | AEFH | ···K |
| push (L) | BCDGI | AEFHL | ···K |
| pop () | BCDGIL | AEFH | ···K |
| deq () | BCDGILK | AEFH | ···· |
| enq (M) | BCDGILK | AEFH | ····M |

Q4 contd..

| operations | output | Stack | Queue |
|---|---|---|---|
| enq(N) | BCDGILK | AEFH | ....M |
| enq (O) | BCDGILK | AEFH | ....M |
| pop () | BCDGILKH | AEF | ....M |
| deq () | BCDGILKHM | AEF | ..... |

10 Marks for complete correct Answer.

Otherwise Partial Marks based on percentage of correctness

---

Q5 (A)

What operation is performed by function (Q) ?

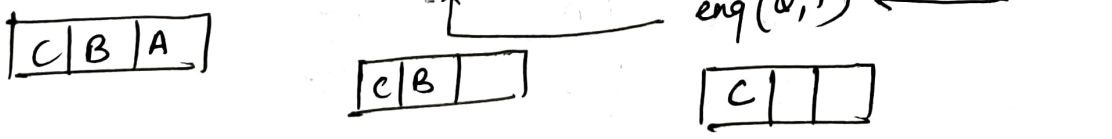
⇒ This function reverse the elements present in Queue [2 Marks].

Justification

⇒ Let 'Q' has 3 elements : A B C [3 Marks].

first call

i = A
function (Q) ⟹ i=B
enqueue(Q, i)
function (Q) ⟹ i = C
enq(Q, i)
function (Q) ⟹ Q is empty
enq(Q, i)




---

## Q5 (B)

```
for (int i=1 ; i<=n ; i++)
    for (int j=i ; j<=n ; j++)
        printf ("%d", i+j);
```

for $i=1 \Rightarrow j = 1$ to $n \Longrightarrow n$ times.

$i=2 \Rightarrow j = 2$ to $n \Longrightarrow n-1$ times

$i=3 \Rightarrow j = 3$ to $n \Longrightarrow n-2$ times

⋮

$i=n \Longrightarrow j = n$ to $n \Longrightarrow 1$ times.

Computation steps

Complexity $C(n) = n + (n-1) + (n-2) + \cdots + 3 + 2 + 1$ [3 Marks]

$= 1 + 2 + 3 + \cdots + n$

$= \frac{n(n+1)}{2} = \underline{O(n^2)}$ [2 Marks for final answer]

## Q5 (C)

```
while (n>=2)
    n= sqrt (n);
```

Assume 'n' to be power of 2.

Initially $n$

1st iteration $n^{th} = n^{1/2^1}$

2nd iteration $n^{1/4} = n^{1/2^2}$

⋮

$i^{th}$ iteration $\boxed{n^{1/2^i} = 2}$

[3 Marks for computation steps.]

Taking log on both sides.

$\frac{1}{2^i} \log n = \log 2 = 1$

$2^i = \log n$

$\boxed{i = \log(\log n)}$ → Taking log on both sides

$C(n) = O(\log \log n)$ [2 marks] for answer.