

DSA Lab Exam Practice Questions

Array

Q1. Find Second Largest Element

Problem: Write a program to find the second largest element in an array.

Input:

- First line: integer n (size of the array)
- Second line: n space-separated integers

Output:

- Single integer representing the second largest element. If not possible, print -1.

Example:

Input:

5

10 20 4 45 99

Output:

45

Array

Q2. Reverse an Array

Problem: Reverse the given array in-place.

Input:

- First line: integer n
- Second line: n space-separated integers

Output:

- Reversed array elements separated by space.

Example:

Input:

5

1 2 3 4 5

Output:

5 4 3 2 1

Linked List

Q3. Insert Node at Specific Position

Problem: Create a singly linked list and insert a new node with a given value at a specified position.

Input:

- First line: integer n (number of nodes)
- Second line: n space-separated integers (node data)
- Third line: integer pos (1-based index position)
- Fourth line: integer val (data of new node)

Output:

- Print the linked list after insertion.

Example:

Input:

```
4
1 2 3 4
3
99
```

Output:

```
1 2 99 3 4
```

Linked List

Q4. Delete Node by Value

Problem: Delete the first occurrence of a given value from a linked list.

Input:

- First line: integer n
- Second line: n space-separated integers
- Third line: integer x (value to delete)

Output:

- Print the linked list after deletion. If value not found, print "Not Found".

Example:

Input:

```
5
10 20 30 40 50
30
```

Output:

```
10 20 40 50
```

Stack

Q5. Balanced Parentheses

Problem: Check if the given expression has balanced parentheses using stack.

Input:

- A string containing parentheses: ()[]{}

Output:

- Print "Balanced" or "Not Balanced".

Example:

Input:

{{[]}}

Output:

Balanced

Stack

Q6. Evaluate Postfix Expression

Problem: Evaluate a postfix expression using stack.

Input:

- A single line containing a postfix expression with space-separated tokens.

Output:

- Integer result of the expression.

Example:

Input:

5 6 2 + * 12 4 / -

Output:

37

Queue

Q7. Implement Circular Queue

Problem: Implement a circular queue with the following operations: ENQUEUE, DEQUEUE, DISPLAY.

Input:

- First line: integer n (size of queue)

- Next lines: commands (ENQUEUE x, DEQUEUE, DISPLAY, STOP)

Output:

- Output after each DISPLAY command.

Example:

Input:

5

ENQUEUE 10

ENQUEUE 20

DEQUEUE

DISPLAY

STOP

Output:

20

Queue

Q8. Generate Binary Numbers from 1 to N using Queue

Problem: Given a number N, print binary numbers from 1 to N using a queue.

Input:

- Integer N

Output:

- N binary numbers separated by space.

Example:

Input:

5

Output:

1 10 11 100 101

Recursion

Q9. Sum of Digits using Recursion

Problem: Find the sum of digits of a number using recursion.

Input:

- Single integer n

Output:

- Sum of digits

Example:

Input:

12345

Output:

15

Recursion

Q10. Tower of Hanoi

Problem: Solve the Tower of Hanoi problem and print the steps.

Input:

- Integer n (number of disks)

Output:

- Steps showing movement from source to destination.

Example:

Input:

3

Output:

Move disk 1 from A to C

Move disk 2 from A to B

Move disk 1 from C to B

Move disk 3 from A to C

Move disk 1 from B to A

Move disk 2 from B to C

Move disk 1 from A to C

Searching Algorithms

Q11. Linear Search

Problem: Find the position of a given element using Linear Search.

Input:

- First line: integer n
- Second line: n space-separated integers
- Third line: integer key

Output:

- Index (1-based) if found, else -1.

Example:

Input:

5

10 20 30 40 50

30

Output:

3

Searching Algorithms

Q12. Binary Search

Problem: Perform Binary Search on a sorted array.

Input:

- First line: integer n
- Second line: n space-separated integers (sorted)
- Third line: integer key

Output:

- Index (1-based) if found, else -1.

Example:

Input:

6

5 10 15 20 25 30

25

Output:

5

Sorting Algorithms

Q13. Bubble Sort

Problem: Sort an array using Bubble Sort algorithm.

Input:

- First line: integer n

- Second line: n space-separated integers

Output:

- Sorted array elements separated by space.

Example:

Input:

5

64 34 25 12 22

Output:

12 22 25 34 64

Q14. Insertion Sort

Problem: Sort an array using Insertion Sort algorithm.

Input:

- First line: integer n

- Second line: n space-separated integers

Output:

- Sorted array elements separated by space.

Example:

Input:

6

12 11 13 5 6 7

Output:

5 6 7 11 12 13

Q15. Merge Sort

Problem: Implement Merge Sort and print the sorted array.

Input:

- First line: integer n
- Second line: n space-separated integers

Output:

- Sorted array elements separated by space.

Example:

Input:

5

38 27 43 3 9

Output:

3 9 27 38 43