

Data Structure Question Bank

Chapter 1: Pointers

1. Define Pointer.
2. Explain the creation of a pointer with an example
3. State the advantages & disadvantages of pointers.
4. How is pointer declared? Give Example.
5. What is a pointer to a pointer?
6. List the operators used with the pointer & state its uses.
7. Write a program to demonstrate the accessing of elements of an array using its base address
8. Program to check the number is EVEN/ODD or +ve/-ve using pointers.

Chapter 2: Dynamic Memory Allocation

1. What is calloc function? (malloc, realloc or free functions can also be given)
2. Differentiate between:
 - a. Calloc and malloc
 - b. Calloc and realloc
 - c. Malloc and free
3. State the difference between static & dynamic memory allocation.
4. Explain the following functions with syntax & example:
a] malloc() b] calloc() c] realloc() d] free()
5. State the difference between malloc() & calloc().
6. State the use of mem() with example.

Chapter 3: Files

1. Define file.
2. Explain file declaration with syntax & example.
3. Explain fopen() & fclose() with syntax & example.
4. What are the various File Access Modes?
5. Explain file input functions with syntax & example.
OR
Explain fread(), fscanf(), fgetc(), fgets() with syntax and example.
6. Explain file write functions with syntax & example.
Or
Explain fwrite(), fprintf(), fputc(), fputs() with syntax and example.
7. Explain fseek(), ftell() & rewind(), feof(), ferror() with syntax & example.
8. Program to generate employee pay slip using files.

Chapter 4: Introduction to Data structures

1. Define Data structures.

2. Explain the classification of Data structure.
3. What is primitive & non-primitive Data structure?
4. List the operation on data structures.

Chapter 5: SEARCHING AND SORTING

1. Differentiate between linear and binary search.
2. What are the advantages of binary search over linear search?
3. Give the analysis of bubble sort. (any other sorting algorithm can be given)
4. Write a program to implement linear search
5. List the advantages and disadvantages of (any one sorting) method
6. Compare Insertion and Selection sort
7. Write C program to implement linear search / binary search.
8. Write a C program to implement binary search recursively.
9. Write a C program to sort an array using bubble sort/quick sort/selection sort/heap sort/merge sort/insertion sort/selection sort.
10. Explain bubble sort/quick sort/selection sort/heap sort/merge sort/insertion sort/selection sort to sort the given list:
11. 88,33,11,44,22,77,66
12. Explain merge sort and apply it on the following list of numbers:
13. 81,32,21,34,22,37,66
14. Explain (any one sorting technique) with an example

Chapter 6: Recursion

1. What is recursion? Explain with an example.
2. Explain types of recursion with example.
3. What are the advantages & disadvantages of recursion?
4. Compare and contrast recursion & iteration.
5. Write a program to find the factorial of a number using recursive function.
6. Write a program to find nth Fibonacci number using recursive function.
7. Write a program to find GCD of 2 numbers using recursive function.
8. Write a program to find the binomial co-efficient using recursive function.
9. Write a program to implement Towers of Hanoi using recursive function.

Chapter 7: Stacks

1. What is Stack?
2. Write the applications of a stack.
3. Define push and pop operations.
4. List & explain the operations performed on stack?
5. How stack is checked for overflow & under flow explain with syntax and example?
6. Explain stack creation with syntax & example.
7. Write an algorithm for PUSH (), POP () & DISPLAY () operations in stack.

8. What are the applications of stack?
9. Define postfix, prefix and infix notations. (any one for one mark)
10. Write the prefix form of $(A+B)/(X-Y)$
11. What is arithmetic expression? Explain the different types of arithmetic expression with example.
12. Problems on conversion of arithmetic expressions from one form to another.
Convert the following to its equivalent postfix and prefix notation:
 - a. $(A+B)+(C/D)*E$
 - b. $A+B*(C+D-E)*F$
 - c. $A+(b*C-D/E)*F$
 - d. $(A-B)/(P+Q)$
 - e. $A+B^C+D*E-A*C$
 - f. $(A*B/C+E*F^2)^4+D$
13. Problems on evaluation of postfix expression using stack.
 - a. $5\ 3\ +\ 2\ *\ 6\ 9\ 7\ -\ /\ -$
 - b. $3\ 1\ +\ 2\ ^\ 7\ 4\ -\ 3\ *\ +\ 5\ -$
 - c. $4\ 5\ +\ 6\ 4\ -\ *\ 4\ 1\ -\ 2\ ^\ +$
 - d. $A\ B\ +\ C\ -\ B\ A\ +\ C\ ^\ -$ when $A=1, B=2, C=3$
 - e. $A\ B\ C\ +\ *\ C\ B\ A\ -\ +\ *$ when $A=1, B=2, C=3$
14. Algorithm to convert infix expression to postfix.
15. Program to implement stack.
16. Program to evaluate postfix expression using stack.

Chapter 8: Queue

1. What is queue?
2. Write the applications of a queue
3. Explain the different types of queues.
4. What is the disadvantage of a linear queue?
5. Define circular queue. Where is it used?
6. State the difference between stacks and queues.
7. What is simple queue? Explain the various operations performed on it.
8. Write an algorithm for Qinsert(), Qdelete() & Qdisplay().
9. How is Queue checked for overflow & under flow explain with syntax and example?
10. Explain queue creation with syntax & example.
11. Write an algorithm for CQinsert(), CQdelete() & CQdisplay().
12. What is a deque? What are the different types of dequeues?
13. Explain priority queue. What are the different types of priority queues?
14. How is CQueue checked for overflow & under flow? Explain with syntax and example?
15. State the advantages of simple queue over circular queue.
16. Program to implement simple/ordinary queue.

17. What is the drawback of a linear queue? How can this drawback be eliminated? Explain.
18. Program to implement circular queue.

Chapter 9: Linked List

1. What is linked list?
2. State the difference between array and linked list
3. Explain different types of linked list with neat diagram.
4. List the operations performed on linked list.
5. State the advantages of:
 - a. Circular singly linked list over singly linked list.
 - b. Doubly linked list over singly linked list.
 - c. Circular Doubly linked list over singly linked list
6. Program to create linked list and perform the following:
 - a. Insert node at the end of the list.
 - b. Delete the node from start of the list.
 - c. Display the values from all the nodes.
7. Program to implement stack using linked list.
8. Program to implement queue using linked list