

Part I - Clarity and neatness - 2 marks

Part II Answer any 9 of the following - 2 marks each

1. #include <filename> is processed by the
2. Use the function to allocate memory for n objects of size m each.
3. Calling a function by using the name of a variable for its parameters is called
4. To use a function declared in a different source file use the qualifier.
5. int matrix[5][5] is a 2D array. Define the function signature for Scale() that takes a 2D array and an integer n as parameters and returns nothing.
6. Consider char str[10], show the array layout with indices for the string "Hello" when stored in the variable str.
7. The relationship between the height of a tree and the maximum number of nodes in a binary tree can be expressed as
8. In what order does one add and delete from a Stack?
9. Sorting a list of numbers by finding the minimum value in successive iterations is the algorithm for
10. Sorting algorithms that do not require extra memory in addition to the original array of data items are called algorithms.
11. Consider the statement,
*const char *cptr = "I find that the harder I work, the more luck I seem to have.";*
**(cptr+=25)* dereferences to character.
12. How many times does the loop below execute?
for (int i =0; i < 5; i+=4) {
 if (++i == 4)
 printf("%d\n", i++);
 else i = i << 2;
}

Part III - Answer the following - 5 marks each

13. Given the inorder and preorder traversals of a Binary Tree, construct the tree and write the postorder traversal for the same.
inorder 2-7-5-6-11-10-16-3-4-9
preorder 10-7-2-6-5-11-3-16-9-4
14. Consider the list of numbers 2, 4, 6, 8, 10, 12, 14 from left to right, on which you perform the following
 1. push(), push(), push(), push(), pop(), pop(), push(), push(), pop(), push(), pop(). Illustrate the stack at the end of the operations and label the top of the stack.

Total Marks: 100; Time: 3 Hours

2. enqueue(), enqueue(), enqueue(), enqueue(), dequeue(), dequeue(), enqueue(), enqueue(), dequeue(), enqueue(). Illustrate the queue at the end of the operations and label the front and back of the queue.
15. Every string in C terminates with a '\0' character. Write a function in C to reverse a string recursively.
16. Write a function to recursively search a Binary Search Tree (BST) for a given key value.

Part IV - Answer the following

17. Write a program to implement a Queue using arrays. Follow the step-wise instructions below : - **20 marks**
 1. Write the steps to include necessary header files - **1 mark**
 2. Declare global pointers called Front and Back and initialise them to point to nothing. - **1 mark**
 3. Declare a struct to define the queue - **2 marks**
 4. Declare and define functions *enqueue*, *dequeue*, *getFront*, *getRear*, *IsEmpty* and *isFull* - **10 marks**
 1. Define the function signature of each of the above functions correctly specifying type of parameters and the return type.
 2. Define the function body correctly.
 5. Define a driver main function that calls the correct functions in the queue to do the following: - **6 marks**
 1. Add 20, 30, 40, 50 to the queue
 2. delete 2 numbers from the queue
 3. Display front and back of the queue
 4. Add 60, 70 to the queue
 5. Delete one number from the queue
 6. Display the front and back of the queue
18. Consider the expression $x \wedge (x + y) - z * a * x \wedge 3 - (b + z) / y$. **20 marks**
 1. Deduce the postfix expression showing the stack trace. The associativity of operators +, -, \times is left-to-right and that of operator \wedge is from right-to-left. The precedence of operators (from highest to lowest) is \wedge , $*$, +, -. **10 marks**
 2. Show the stack trace for evaluating the same postfix expression **8 marks**
 3. Substitute values $x = 4$, $y = 3$, $z = 1$, $a = 2$, $b = 5$ and evaluate the resulting postfix expression **2 marks**
19. Sort the array containing 8 18 11 54 99 2 60 7 72 106 using Insertion Sort and show the visualisation of the algorithm. **10 marks**
20. Sort the array containing 46 9 23 12 0 34 2 7 15 89 67 using Selection Sort and show the visualisation of the algorithm. **10 marks**