Reg. No. : | E | N | G | G | T | R | E | E | . | C | O | M |

Question Paper Code : 50899

B.E./B.Tech. DEGREE EXAMINATIONS, APRIL/MAY 2024.

Third Semester

Electrical and Electronics Engineering

CS 3353 — C PROGRAMMING AND DATA STRUCTURES

(Common to Electronics and Communication Engineering/Electronics and Instrumentation Engineering/Electronics and Tele Communication Engineering/Instrumentation and Control Engineering)

(Regulations 2021)

(Also Common to PTCS 3353 – C Programming and Data Structures for B.E. (Part - Time) – Second Semester – Electronics and Communication Engineering – Regulations – 2023)

Time : Three hours | Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. What is recursive function?
2. When is the ternary operator used?
3. State the usage of the union in C.
4. What are preprocessor directives?
5. What are Abstract Data Types (ADT)?
6. State the applications of queue.
7. When is rehashing necessary?
8. What is an expression tree?
9. What are the limitations of linear search?
10. What is the fundamental concept of merge sort?

PART B — (5 × 13 = 65 marks)

11. (a) (i) Why we use functions in C languages. Give example. (6)

(ii) Write both iterative and recursive functions in C to evaluable a^b. (7)

Or

(b) (i) What are the different types of control statements? Explain with example. (6)

(ii) Write a C program that computes the sum of the following series up to n terms. $1-\frac{x^2}{2!}+\frac{x^4}{4!}-\frac{x^6}{6!}+\ldots$ (7)

12. (a) Write C functions to perform the following operations with two-dimensional arrays.

(i) Reading any two dimensional array elements. (3)

(ii) Find the sum of odd and the even array elements. (3)

(iii) Finding maximum and minimum of array elements. (3)

(iv) Printing the transpose. (4)

Or

(b) Explain the structure, nested structure, and self-referential structure with examples.

13. (a) Explain the stack ADT. State and explain the different representation of stack with example.

Or

(b) Explain the procedure for converting an infix expression to a postfix expression using a stack. Convert the following infix expression into a postfix expression :

$x*(w+y/2*x\hat{}(4+x))$ and evaluate the obtained postfix expression using the values $(x=1, y=2, w=3)$.

14. (a) State and explain the array and linked implementations of a binary search tree with pseudo code and an example.

Or

(b) Consider the binary search tree provided in Figure 1. Perform the following operations:

(i) Determine the results of in-order, pre-order and post-order traversals. (4)

(ii) Insert the nodes 12, 22, 33, 44, and 55. (4)

(iii) Demonstrate the deletion of the node 43. (5)

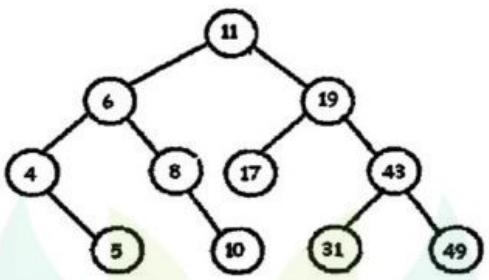

Figure – 1

15. (a) Explain the basic idea behind insertion sort and how it works. Illustrate with pseudo code and a suitable example.

Or

(b) Distinguish between min heap and max heap. Show how heap sort process the input 142, 543, 123, 65, 453, 879, 572, 434, 111, 242, 311 and 102.

PART C — (1 × 15 = 15 marks)

16. (a) Given input $\{4371, 1323, 6173, 4199, 4344, 9679, 1989\}$ and a hash function $h(x) = X \pmod{10}$, show the resulting

(i) Separate chaining hash table. (3)

(ii) Open addressing hash table using linear probing. (4)

(iii) Open addressing hash table using quadratic probing. (4)

(iv) Open addressing hash table with second hash function $h2(x) = 7 - (x \bmod 7)$. (4)

Or

(b) Sort the sequence 4, 6, 8, 2, 9, 5, 1, 7 and 3 using the following

(i) Merge sort (8)

(ii) Quick sort (Picking the first element as the pivot). (7)

———————