

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Lab Test/Exam Programs List Subject: Data Structures (22CS36L)

1. a) Define a structure called Student with the members: Name, Reg_no, marks in 3 tests and average_marks.
Develop a menu driven program to perform the following by writing separate function for each operation: a) read information of N students b) display student's information c) to calculate the average of best two test marks of each student.
Note: Allocate memory dynamically and illustrate the use of pointer to an array of structure.
b) Define a structure called Time containing 3 integer members (Hour, Minute, Second). Develop a menu driven program to perform the following by writing separate function for each operation.
a) Read (T) :To read time b) Display (T):To display time c) update(T):To Update time d) Add(T1, T2) : Add two time variables.
Update function increments the time by one second and returns the new time (if the increment results in 60 seconds, then the second member is set to zero and minute member is incremented by one. If the result is 60 minutes, the minute member is set to zero and the hour member is incremented by one. Finally, when the hour becomes 24, Time should be reset to zero. While adding two time variables, normalize the resultant time value as in the case of update function. Note: Illustrate the use of pointer to pass time variable to different functions.
2. Develop a menu driven program to implement the following operations on an array of integers with dynamic memory allocation. Display the array contents after each operation.
 - i) Insert by position ii) Delete by key iii) Search by position iv) Reverse the contents.
3. Develop a menu driven program to implement the following operations on an array of integers with dynamic memory allocation. Display the array contents after each operation.
 - i) Insert by order ii) Delete by position iii) Search by key iv) Reverse the contents.
4. Implement circular single linked list to perform the following operations
 - i) Insert by order ii) Delete by position iii) Search for an item by key iv) Delete by keyDisplay the list contents after each operation
5. Implement circular double linked list to perform the following operations
 - i) Insert by order ii) Delete by position iii) Delete by key iv) Search by positionDisplay the list contents after each operation

6. Implement circular single linked list to perform the following operations
 - i) Insert front ii) Insert rear iii) Delete a node with the given key iv) Search for an item by position

Display the list contents after each operation

7. Implement circular double linked list to perform the following operations

- i) Insert front ii) Insert rear iii) Delete by position iv) Search by key
- Display the list contents after each operation

8. Implement circular single linked list to perform the following operations

- i) Insert by position ii) Delete rear iii) Delete Front iv) Search for an item by value
- Display the list contents after each operation

9. Implement circular double linked list to perform the following operations

- i) Insert by order ii) Delete rear iii) Delete Front iv) Search for an item by position
- Display the list contents after each operation

10. Develop a menu driven program to convert infix expression to postfix expression using stack and evaluate the postfix expression. (Test for nested parenthesized expressions) **Note: Define Stack structure and implement the operations. Use different stacks for conversion and evaluation.**

11. Develop a menu driven program to convert infix expression to prefix expression using stack and evaluate the prefix expression (Test for nested parenthesized expressions)

Note: Define Stack structure and implement the operations. Use different stacks for conversion and evaluation.

12. Develop a menu driven program to implement the following types of Queues by allocating memory dynamically.

- i) Circular Queue ii) Double ended Queue

Note: Define Queue structure and implement the operation with array representation

13. Develop a menu driven program to implement the following types of Queues by allocating memory dynamically.

- i) Circular Queue ii) Priority Queue

Note: Define Queue structure and implement the operation with array representation

14. Develop a menu driven program to implement Binary Search tree with the following operations.

- i) Construction ii) Traversals (Pre, In and Post Order) iii) Searching a node by key and displaying its information along with its parent if exists, otherwise a suitable message. iv) Counting all nodes. v) Finding height. vi) Finding node with the Maximum key value and printing the node details along with the parent.

15. Develop a menu driven program to implement Binary Search tree with the following operations.

- i) Construction ii) Traversals (Pre, In and Post Order) iii) Searching a node by key and deleting if exists (node to be deleted may be leaf or non-leaf with one child or two children)

Note: For Lab Test : Program 1 to 13 : For Final Lab Exam: Program 1 to 15