

Searching

Q1. Create a random array of n integers. Accept a value x from user and **use linear search** algorithm to check whether the number is present in the array or not and output the position if the number is present. [10]

Q1. Create a random array of n integers. Accept a value x from user and **use linear search algorithm** to check whether the number is present in the array or not and output the position if the number is present. [10]

Q1. Read the data from file 'cities.txt' containing names of cities and their STD codes. Accept a name of the city from user and use linear search algorithm to check whether the name is present in the file and output the STD code, otherwise output "city not in the list". [10]

Q1. Read the data from file 'cities.txt' containing names of cities and their STD codes. Accept a name of the city from user and use sentinel linear search algorithm to check whether the name is present in the file and output the STD code, otherwise output "city not in the list". [10]

Q2. Read the data from file 'cities.txt' containing names of cities and their STD codes. Accept a name of the city from user and use sentinel linear search algorithm to check whether the name is present in the file and output the STD code, otherwise output "city not in the list". [20]

Sorting

Q1. Read the 'n' numbers from user and sort using bubble sort. [10]

Q1. Sort a random array of n integers (accept the value of n from user) in ascending order by using **selection sort algorithm**. [10]

Q1. Sort a random array of n integers (accept the value of n from user) in ascending order by using **selection sort algorithm**. [10]

Q1. Sort a random array of n integers (accept the value of n from user) in ascending order by using **selection sort algorithm** [10]

Q1. Sort a random array of n integers (accept the value of n from user) in ascending order by using insertion sort algorithm. [10]

Q1. Sort a random array of n integers (accept the value of n from user) in ascending order by using quick sort algorithm. [10]

Q2. Read the data from the file "employee.txt" and sort on names in alphabetical order (use strcmp) using insertion sort and selection sort. [20]

Q2. Read the data from the 'employee.txt' file and sort on age using Counting sort, Merge sort, Quick sort and write the sorted data to another file 'sortedemponage.txt'. [20]

Q2. Read the data from the file "employee.txt" and sort on names in alphabetical order (use strcmp) using bubble sort and selection sort. [20]

Q2. Read the data from the 'employee.txt' file and sort on age using Merge sort, Quick sort and write the sorted data to another file 'sortedemponage.txt'. [20]

Q2. Read the data from the file and sort on names in alphabetical order (use strcmp) using Counting sort, Merge sort and write the sorted data to another file 'sortedemponame.txt' [20]

Q2 Read the data from the 'employee.txt' file and sort on age using Counting sort and Quick sort and write the sorted data to another file 'sortedemponage.txt'. [20]

Q1. Sort a random array of n integers (accept the value of n from user) in ascending order by using recursive Counting sort algorithm. [10]

Q2. Read the data from the file "employee.txt" and sort on names in alphabetical order (use strcmp) using insertion sort and selection sort [20]

Stack

Q1. Write a program that reverses a string of characters. The function should use a stack library(cststack.h). Use a static implementation of the stack. [10]

Q1. Implement a stack library (ststack.h) of integers using a static implementation of the stack and implementing the operations like init(S), S=Push(S,x) and isEmpty(S). Write a driver program that includes stack library and calls different stack operations. [10]

Q1. Implement a stack library (ststack.h) of integers using a static implementation of the stack and implementing the above six operations. Write a driver program that includes stack library and calls different stack operations. [10]

Q1. Implement a stack library (ststack.h) of integers using a static implementation of the stack and implementing the operations like init(S), S=push(S), and X=peek(S). Write a driver program that includes stack library and calls different stack operations. [10]

Q1. Implement a stack library (ststack.h) of integers using a static implementation of the stack and implementing the operations like init(S), S=push(S) and S=pop(S). Write a driver program that includes stack library and calls different stack operations. [10]

Q1. Implement a stack library (ststack.h) of integers using a static implementation of the stack and implementing the operations like init(S), S=push(S), isFull(S). Write a driver program that includes stack library and calls different stack operations. [10]

Q1. Write a program to convert an infix expression of the form $(a*(b+c)*((da)/b))$ into its equivalent postfix notation. Consider usual precedence's of operators. Use stack library of stack of characters using static implementation. [10]

Q1. Write a program to convert an infix expression of the form $(a*(b+c)*((da)/b))$ into its equivalent postfix notation. Consider usual precedence's of operators. Use stack library of stack of characters using static implementation. [10]

Q2. Write a program that checks whether a string of characters is palindrome or not. The function should use a stack library (cststack.h) of stack of characters using a static implementation of the stack. [20]

Q2. Write a C program to check whether the contents of two stacks are identical. Use stack library to perform basic stack operations. Neither stack should be changed. [20]

Q2. A postfix expression of the form $ab+cd-*ab/$ is to be evaluated after accepting the values of a, b, c and d. The value should be accepted only once and the same value is to be used for repeated occurrence of same symbol in the expression. Formulate the problem and write a C program to solve the problem by using stack [20]

Q2 Write a program that copies the contents of one stack into another. Use stack library to perform basic stack operations. The order of two stacks must be identical.(Hint: Use a temporary stack to preserve the order). [20]

Q2. Write a program that copies the contents of one stack into another. Use stack library to perform basic stack operations. The order of two stacks must be identical.(Hint: Use a temporary stack to preserve the order). [20]

Q2. Write a C program to evaluate postfix expression. [20]

Queue

Q2. Write a program to reverse the elements of a queue using queue library. Implement basic queue operations init, enqueue, dequeue. [20]

Q2. Implement a priority queue library (PriorityQ.h) of integers using a static implementation of the queue and implement the below two operations. 1) Add an element with its priority into the queue. 2) Delete an element from queue according to its priority. [20]

Q2. Implement a queue library (dyqueue.h) of integers using a dynamic (linked list) implementation of the queue and implement init, enqueue, dequeue, isempty, peek operations. [20]

Q1. Implement a linear queue library (st_queue.h) of integers using a static implementation of the queue and implementing the init(Q), add(Q) and peek(Q) operations. Write a program that includes queue library and calls different queue operations. [10]

Q1. Implement a circular queue library (cir_queue.h) of integers using a dynamic (circular linked list) implementation of the queue and implementing the operations like init (Q), AddQueue(Q, x) and isEmpty (Q). Write a menu driven program that includes queue library and calls different queue operations. [10]

Q1. Implement a linear queue library (st_queue.h) of integers using a static implementation of the queue and implementing the operations like init (Q), AddQueue(Q, x) and X=DeleteQueue(Q). Write a program that includes queue library and calls different queue operations. [10]

Q2. Read the data from the file "employee.txt" and sort on names in alphabetical order (use strcmp) using bubble sort and selection sort. [20]

Q2. Implement a priority queue library (PriorityQ.h) of integers using a static implementation of the queue and implementing the below two operations. Write a driver program that includes queue library and calls different queue operations. 1) Add an element with its priority into the queue. 2) Delete an element from queue according to its priority. [20]

Q2. Implement a circular queue library (cir_queue.h) of integers using a dynamic (circular linked list) implementation of the queue and implementing init(Q), AddQueue(Q) and DeleteQueue(Q) operations. Write a menu driven program that includes queue library and calls different queue operations. [20]

Q2. A doubly ended queue allows additions and deletions from both the ends that is front and rear. Initially additions from the front will not be possible. To avoid this situation, the array can be treated as if it were circular. Implement a queue library (dstqueue.h) of integers using a static implementation of the circular queue and implementing the following operations. a. isFull(Q) b. addFront(Q) c. getRear(Q) d. deleteRear(Q) [20]

Q2. Implement a linear queue library (dyqueue.h) of integers using a dynamic (circular linked list) implementation of the queue and implementing the five queue operations (init(Q), AddQueue(Q, x), X=DeleteQueue(Q), X=peek(Q), isEmpty(Q)). [20] Write a program to reverse the elements of a queue using queue library.

Q1. Implement a priority queue library (PriorityQ.h) of integers using a static implementation of the queue and implementing the below operation [10] Add an element with its priority into the queue Q2. Read the data from file 'sortedcities.txt' containing sorted names of cities and their STD codes. Accept a name of the city from user and use binary search algorithm to check whether the name is present in the file and output the STD code, otherwise output "city not in the list". [20]

Linked List

Q1. Implement a list library (doublylist.h) for a doubly linked list of integers with the create, display operations. Write a menu driven program to call these operations. [10]

Q2. Write a program that sorts the elements of linked list using bubble sort technique. [20]

Q2. Write a program that sorts the elements of linked list using any of sorting technique. [20]

Q1. Implement a list library (singlylist.h) for a singly linked list of integer with the operations create, display. Write a menu driven program to call these operations [10]

Q1. Implement a list library (singlylist.h) for a singly linked list of integer With the operations create, delete specific element and display. Write a menu driven program to call these operations [10]

Q2. There are lists where insertion should ensure the ordering of data elements. Since the elements are in ascending order the search can terminate once equal or greater element is found. Implement a singly linked list of ordered integers (ascending/descending) with insert, search, and display operations. [20]

Q1. Write a program that adds two single variable polynomials. Each polynomial should be represented as a list with linked list implementation. [10]

Q2. There are lists where new elements are always appended at the end of the list. The list can be implemented as a circular list with the external pointer pointing to the last element of the list. Implement singly linked circular list of integers with append and display operations. The operation append(L, n), appends to the end of the list, n integers accepted from user. [20]

Q1. Write a program that merges two ordered linked lists into third new list. When two lists are merged the data in the resulting list are also ordered. The two original lists should be left unchanged. That is merged list should be new one. Use linked implementation. [10]

Q1 Implement a list library (singlylist.h) for a singly linked list. Create a linked list, reverse it and display reversed linked list. [10]

Q1. Sort a random array of n integers (accept the value of n from user) in ascending order by using recursive Counting sort algorithm [10]

Q2. Write a program that multiply two single variable polynomials. Each polynomial should be represented as a list with linked list implementation [20]

Q2. There are lists where insertion should ensure the ordering of data elements. Since the elements are in ascending order the search can terminate once equal or greater element is found. Implement a doubly linked list of ordered integers (ascending/descending) with insert, search and display operations. [20]

Q2. There are lists where new elements are always appended at the end of the list. The list can be implemented as a circular list with the external pointer pointing to the last element of the list. Implement singly linked circular list of integers with append and display operations. The operation `append(L, n)`, appends to the end of the list, n integers either accepted from user or randomly generated. [20]