

1. Write a program to accept 'n' numbers into an array and then calculate the sum of numbers present in odd positions and even positions respectively.

Output:

Enter how many numbers: 4

Enter 4 numbers:

23 45 2 3

Sum of numbers at even positions = 25

Sum of numbers at odd positions = 48

1B. Write a program to implement a Binary Search algorithm. Write a search function which takes a SearchList as its first parameter and a Comparable as its second. If either parameter is null, or if the SearchList is empty, you should return NULL. implement the following algorithm:

- Examine the value in the middle of the current array and print it.
- If the midpoint value is the value that we are looking for, return true
- If the value that we are looking for is greater than the midpoint value, adjust the current array to start at the midpoint and print the index.
- if the value that we are looking for is less than the midpoint value, adjust the current array to end at the midpoint and print the index.
- Continue until you find the value, or until the start reaches the end

Output:

Enter how many elements: 4

Enter 4 elements in sorted order:

34 56 67 89

Enter element to search: 56

Checking middle value: 56 at index 1

Element found at index 1.

1c. Write a program which creates a structure Student which must have the attribute Avg Marks with 3 more attributes sort the list of the student in descending order using Bubble sort.

Output :

Enter details of student 1:

Roll No: 4

Name: prachi

Average Marks: 45

Enter details of student 2:

Roll No:5

Name: Prajakta

Average Marks:65

2.Design, Develop and Implement a menu driven Program in C for the following operations on STACK of Integers (Array Implementation of Stack with maximum size MAX) a. Push an Element on to Stack b. Pop an Element from Stack c. Demonstrate how Stack can be used to check Palindrome d. Demonstrate Overflow and Underflow situations on Stack e. Display the status of Stack f. Exit Support the program with appropriate functions for each of the above operations

Output:

--- Stack Menu ---

1. Push
2. Pop
3. Check Palindrome
4. Display Stack
5. Exit

Enter your choice: 1

Enter value to push: 10

10 pushed into stack.

Enter your choice: 1

Enter value to push: 20

20 pushed into stack.

Enter your choice: 4

Stack elements are:

20

10

Enter your choice: 2

20 popped from stack.

Enter your choice: 3

Enter a string to check palindrome: madam

String is Palindrome.

Enter your choice: 5

Exiting...

3.Design, Develop and Implement a menu driven Program in C for the following operations on QUEUE of Characters (Array Implementation of Queue with maximum size MAX) a. Insert an Element on to QUEUE b. Delete an Element from QUEUE c. Demonstrate Overflow and Underflow situations on QUEUE d. Display **the status of QUEUE e. Exit Support the program with appropriate functions** **for** each of the above operations

Output:

--- Queue Menu ---

1. Enqueue

2. Dequeue

3. Display

4. Exit

Enter your choice: 1

Enter character to insert: A

A inserted into queue.

Enter your choice: 1

Enter character to insert: B

B inserted into queue.

Enter your choice: 3

Queue elements are: A B

Enter your choice: 2

A deleted from queue.

Enter your choice: 3

Queue elements are: B

Enter your choice: 4

Exiting...

4.This C program creates a linked list to store integer elements. It prompts the user to enter elements and add them to the list until the user enters 0. It then traverses the list and prints each element and "=>" until reaching the null pointer. Finally, it displays the number of nodes in the list

Output:

Enter integer elements to add to the linked list (enter 0 to stop):

10 20 30 40 0

Linked List: 10 -> 20 -> 30 -> 40 -> NULL

Total number of nodes = 4

5. Write a program to insert Number of Nodes in the Binary Tree and Traverse in Inorder , Preorder and Post Order and Search an element in Binary Tree(Display NULL if not found, If found Display Found)

Output:

Enter number of nodes to insert in the Binary Tree: 5

Enter 5 integer values:

50 30 70 20 40

Inorder Traversal: 20 30 40 50 70

Preorder Traversal: 50 30 20 40 70

Postorder Traversal: 20 40 30 70 50

Enter element to search in the Binary Tree: 40

Found (Element 40 exists in the tree)

6. Implement Binary search tree(BST) with following Menu operations. 1. Search an element in BST(Display NULL if not found, If found Display Found) 2. Insert an element in BST 3. Delete leaf element in BST 4. Exit

Output:

--- Binary Search Tree Menu ---

1. Search an element
2. Insert an element
3. Delete a leaf element
4. Exit

Enter your choice: 2

Enter element to insert: 50

Inorder traversal: 50

--- Binary Search Tree Menu ---

1. Search an element
2. Insert an element
3. Delete a leaf element
4. Exit

Enter your choice: 2

Enter element to insert: 30

Inorder traversal: 30 50

--- Binary Search Tree Menu ---

1. Search an element
2. Insert an element
3. Delete a leaf element
4. Exit

Enter your choice: 2

Enter element to insert: 70

Inorder traversal: 30 50 70

--- Binary Search Tree Menu ---

1. Search an element
2. Insert an element
3. Delete a leaf element
4. Exit

Enter your choice: 1

Enter element to search: 70

Element found in BST.

--- Binary Search Tree Menu ---

1. Search an element
2. Insert an element

3. Delete a leaf element

4. Exit

Enter your choice: 3

Enter leaf element to delete: 30

Leaf node 30 deleted successfully.

Inorder traversal: 50 70

--- Binary Search Tree Menu ---

1. Search an element

2. Insert an element

3. Delete a leaf element

4. Exit

Enter your choice: 3

Enter leaf element to delete: 50

Element 50 is not a leaf node or not found.

Inorder traversal: 50 70

--- Binary Search Tree Menu ---

1. Search an element

2. Insert an element

3. Delete a leaf element

4. Exit

Enter your choice: 4

Exiting program.