

File Edit Selection View Go Run ... Daa-code

screenrec

EXPLORER

- ✓ DAA-CODE
 - > .vscode
 - > daacode-java
 - ≡ a.exe
 - BF_string_matching.cpp
 - BinarySearch.cpp
 - BinarySearchRecursive.cpp
 - BubbleSort.cpp
 - exhaustive_search.cpp
 - HalfLinearSearch.cpp
 - InsertionSort.cpp
 - LinearSearch.cpp
 - LinearSearch.exe
 - MergeSort.cpp
 - QuickSort.cpp
 - SelectionSort.cpp
 - StackUsingArray.cpp
 - StrassenseMatrixMultiplication....
 - travelling_salesman_problem.cpp
- > OUTLINE
- > TIMELINE

StackUsingArray.cpp •

```
StackUsingArray.cpp > push(int)
1 //Name: Danish Maniyar, Reg.no:2020BIT028
2 //stack using array
3 #include <bits/stdc++.h>
4 using namespace std;
5 #define MAX 1000
6 class Stack {
7     int top;
8 public:
9     int a[MAX];
10    Stack() { top = -1; }
11    bool push(int x);
12    int pop();
13    int peek();
14    bool isEmpty();
15 };
16 bool Stack::push(int x){
17     if (top >= (MAX - 1)) {
18         cout << "Stack Overflow";
19         return false;
20     }
21     else {
22         a[++top] = x;
23         cout << x << " pushed into stack\n";
24         return true;
25     }
26 }
27 int Stack::pop(){
28     if (top < 0) {
29         cout << "Stack Underflow";
30         return 0;
31     }
32     else {
```

Ln 16, Col 24 Spaces: 4 UTF-8 CRLF C++ Win32

Type here to search

23°C Clear

ENG IN 12:55 AM 2/21/2023

File Edit Selection View Go Run ...

Daa-code

screenrec

EXPLORER

DAA-CODE

.vscode

daacode-java

a.exe

BF_string_matching.cpp

BinarySearch.cpp

BinarySearchRecursive.cpp

BubbleSort.cpp

exhaustive_search.cpp

HalfLinearSearch.cpp

InsertionSort.cpp

LinearSearch.cpp

LinearSearch.exe

MergeSort.cpp

QuickSort.cpp

SelectionSort.cpp

StackUsingArray.cpp

StrassenseMatrixMultiplication....

travelling_salesman_problem.cpp

OUTLINE

TIMELINE

StackUsingArray.cpp

StackUsingArray.cpp > push(int)

```
32     else {
33         int x = a[top--];
34         return x;
35     }
36 }
37 int Stack::peek(){
38     if (top < 0) {
39         cout << "Stack is Empty";
40         return 0;
41     }
42     else {
43         int x = a[top];
44         return x;
45     }
46 }
47 bool Stack::isEmpty(){
48     return (top < 0);
49 }
50 int main(){
51     class Stack s;
52     s.push(10);
53     s.push(20);
54     s.push(30);
55     cout << s.pop() << " Popped from stack\n";
56     cout << "Top element is : " << s.peek() << endl;
57     cout << "Elements present in stack : ";
58     while(!s.isEmpty()){
59         cout << s.peek() << " ";
60         s.pop();
61     }
62     return 0;
63 }
```

Ln 16, Col 24 Spaces: 4 UTF-8 CRLF C++ Win32

Type here to search

23°C Clear ENG IN 12:55 AM 2/21/2023

File Edit Selection View Go Run ...

← →

Daa-code

screenrec

EXPLORER

DAA-CODE

> .vscode

> daacode-java

a.exe

BF_string_matching.cpp

BinarySearch.cpp

BinarySearchRecursive.cpp

BubbleSort.cpp

exhaustive_search.cpp

HalfLinearSearch.cpp

InsertionSort.cpp

LinearSearch.cpp

LinearSearch.exe

MergeSort.cpp

QuickSort.cpp

SelectionSort.cpp

StackUsingArray.cpp

StrassenseMatrixMultiplication....

travelling_salesman_problem.cpp

StackUsingArray.cpp

push(int)

52 s.push(10);

53 s.push(20);

54 s.push(30);

55 cout << s.pop() << " Popped from stack\n";

56 cout << "Top element is : " << s.peak() << endl;

57 cout << "Elements present in stack : ";

58 while(!s.isEmpty()){

59 cout << s.peak() << " ";

60 s.pop();

61 }

62 return 0;

63 }

64 }

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PS D:\DAA\Daa-code> g++ StackUsingArray.cpp

PS D:\DAA\Daa-code> .\a.exe

10 pushed into stack

20 pushed into stack

30 pushed into stack

30 Popped from stack

Top element is : 20

Elements present in stack : 20 10

PS D:\DAA\Daa-code>

powerShell

powerShell

Ln 16, Col 24 Spaces: 4 UTF-8 CRLF C++ Win32

0 0 0

Type here to search

23°C Clear

ENG IN

12:56 AM 2/21/2023

File Edit Selection View Go Run ...

← →

Daa-code

screenrec

EXPLORER

DAA-CODE

.vscode

daacode-java

a.exe

BF_string_matching.cpp

BinarySearch.cpp

BinarySearchRecursive.cpp

BubbleSort.cpp

exhaustive_search.cpp

HalfLinearSearch.cpp

InsertionSort.cpp

LinearSearch.cpp

LinearSearch.exe

MergeSort.cpp

QueueUsingArray.cpp

QuickSort.cpp

SelectionSort.cpp

StackUsingArray.cpp

StrassenseMatrixMultiplication....

travelling_salesman_problem.cpp

OUTLINE

TIMELINE

StackUsingArray.cpp

QueueUsingArray.cpp

QueueUsingArray.cpp > ...

```
1 //Name:Maniyar Danish, Reg.no:2020BIT028
2 //queue using array
3 #include <bits/stdc++.h>
4 using namespace std;
5 struct Queue {
6     int front, rear, capacity;
7     int* queue;
8     Queue(int c){
9         front = rear = 0;
10        capacity = c;
11        queue = new int;
12    }
13    ~Queue() { delete[] queue; }
14    void queueEnqueue(int data){
15        if (capacity == rear) {
16            printf("\nQueue is full\n");
17            return;
18        }
19        else {
20            queue[rear] = data;
21            rear++;
22        }
23        return;
24    }
25    void queueDequeue(){
26        if (front == rear) {
27            printf("\nQueue is empty\n");
28            return;
29        }
30        else {
31            for (int i = 0; i < rear - 1; i++) {
32                queue[i] = queue[i + 1];
33            }
34        }
35    }
36 }
```

Ln 2, Col 20

Spaces: 4

UTF-8

CRLF

C++

Win32

23°C

Clear

ENG IN

1:02 AM

2/21/2023

File Edit Selection View Go Run ...

← →

Daa-code

screenrec

EXPLORER

DAA-CODE

> .vscode

> daacode-java

a.exe

BF_string_matching.cpp

BinarySearch.cpp

BinarySearchRecursive.cpp

BubbleSort.cpp

exhaustive_search.cpp

HalfLinearSearch.cpp

InsertionSort.cpp

LinearSearch.cpp

LinearSearch.exe

MergeSort.cpp

QueueUsingArray.cpp

QuickSort.cpp

SelectionSort.cpp

StackUsingArray.cpp

StrassenseMatrixMultiplication....

travelling_salesman_problem.cpp

OUTLINE

TIMELINE

StackUsingArray.cpp

QueueUsingArray.cpp

QueueUsingArray.cpp > ...

29 }

30 else {

31 for (int i = 0; i < rear - 1; i++) {

32 queue[i] = queue[i + 1];

33 }

34 rear--;

35 }

36 return;

37 }

38 void queueDisplay(){

39 int i;

40 if (front == rear) {

41 printf("\nQueue is Empty\n");

42 return;

43 }

44 for (i = front; i < rear; i++) {

45 printf(" %d <-- ", queue[i]);

46 }

47 return;

48 }

49 void queueFront(){

50 if (front == rear) {

51 printf("\nQueue is Empty\n");

52 return;

53 }

54 printf("\nFront Element is: %d", queue[front]);

55 return;

56 }

57 };

58 int main(void){

59 Queue q(4);

60 q.queueDisplay();

Ln 2, Col 20 Spaces: 4 UTF-8 CRLF C++ Win32

23°C Clear

ENG IN

1:02 AM

2/21/2023

File Edit Selection View Go Run ...

← →

Daa-code

screenrec

EXPLORER

DAA-CODE

> .vscode

> daacode-java

a.exe

BF_string_matching.cpp

BinarySearch.cpp

BinarySearchRecursive.cpp

BubbleSort.cpp

exhaustive_search.cpp

HalfLinearSearch.cpp

InsertionSort.cpp

LinearSearch.cpp

LinearSearch.exe

MergeSort.cpp

QueueUsingArray.cpp

QuickSort.cpp

SelectionSort.cpp

StackUsingArray.cpp

StrassenseMatrixMultiplication....

travelling_salesman_problem.cpp

QueueUsingArray.cpp > ...

```
58 int main(void){
59     Queue q(4);
60     q.queueDisplay();
61     q.queueEnqueue(20);
62     q.queueEnqueue(30);
63     q.queueEnqueue(40);
64     q.queueEnqueue(50);
65     q.queueDisplay();
66     q.queueEnqueue(60);
67     q.queueDisplay();
68     q.queueDequeue();
69     q.queueDequeue();
70     printf("\n\nafter two node deletion\n\n");
71     q.queueDisplay();
72     q.queueFront();
73     return 0;
74 }
75
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\DAA\Daa-code> g++ QueueUsingArray.cpp
PS D:\DAA\Daa-code> .\a.exe

Queue is Empty
20 <-- 30 <-- 40 <-- 50 <--
Queue is full
20 <-- 30 <-- 40 <-- 50 <--

after two node deletion

40 <-- 50 <--
Front Element is: 40
PS D:\DAA\Daa-code>
```

Ln 2, Col 20 Spaces: 4 UTF-8 CRLF C++ Win32

23°C Clear

ENG IN 1:02 AM 2/21/2023

File Edit Selection View Go Run ...

← →

Daa-code

screenrec

EXPLORER

DAA-CODE

> .vscode

> daacode-java

≡ a.exe

BF_string_matching.cpp

BinarySearch.cpp

BinarySearchRecursive.cpp

BubbleSort.cpp

exhaustive_search.cpp

HalfLinearSearch.cpp

InsertionSort.cpp

LinearSearch.cpp

LinearSearch.exe

LinkedList.cpp

MergeSort.cpp

QueueUsingArray.cpp

QuickSort.cpp

SelectionSort.cpp

StackUsingArray.cpp

StrassenseMatrixMultiplication....

travelling_salesman_problem.cpp

> OUTLINE

> TIMELINE

StackUsingArray.cpp

QueueUsingArray.cpp

LinkedList.cpp X

LinkedList.cpp > ...

1 //Name: Maniyar Danish, Reg.no:2020BIT028

2 //LinkedList

3 #include <iostream>

4 using namespace std;

5 class Node {

6 public:

7 int data;

8 Node* next;

9 Node(){

10 data = 0;

11 next = NULL;

12 }

13 Node(int data){

14 this->data = data;

15 this->next = NULL;

16 }

17 };

18 class Linkedlist {

19 Node* head;

20 public:

21 Linkedlist() { head = NULL; }

22 void insertNode(int);

23 void printList();

24 void deleteNode(int);

25 };

26 void Linkedlist::deleteNode(int nodeOffset){

27 Node *temp1 = head, *temp2 = NULL;

28 int ListLen = 0;

29 if (head == NULL) {

30 cout << "List empty." << endl;

31 return;

32 }

Ln 2, Col 14 Spaces: 4 UTF-8 CRLF C++ Win32

23°C Clear

ENG IN

1:07 AM

2/21/2023

File Edit Selection View Go Run ...

← →

Daa-code

screenrec

EXPLORER

DAA-CODE

> .vscode

> daacode-java

≡ a.exe

BF_string_matching.cpp

BinarySearch.cpp

BinarySearchRecursive.cpp

BubbleSort.cpp

exhaustive_search.cpp

HalfLinearSearch.cpp

InsertionSort.cpp

LinearSearch.cpp

LinearSearch.exe

LinkedList.cpp

MergeSort.cpp

QueueUsingArray.cpp

QuickSort.cpp

SelectionSort.cpp

StackUsingArray.cpp

StrassenseMatrixMultiplication....

travelling_salesman_problem.cpp

> OUTLINE

> TIMELINE

StackUsingArray.cpp

QueueUsingArray.cpp

LinkedList.cpp X

LinkedList.cpp > ...

27 Node *temp1 = head, *temp2 = NULL;

28 int ListLen = 0;

29 if (head == NULL) {

30 cout << "List empty." << endl;

31 return;

32 }

33 while (temp1 != NULL) {

34 temp1 = temp1->next;

35 ListLen++;

36 }

37 if (ListLen < nodeOffset) {

38 cout << "Index out of range"

39 << endl;

40 return;

41 }

42 temp1 = head;

43 if (nodeOffset == 1) {

44 head = head->next;

45 delete temp1;

46 return;

47 }

48 while (nodeOffset-- > 1) {

49 temp2 = temp1;

50 temp1 = temp1->next;

51 }

52 temp2->next = temp1->next;

53 delete temp1;

54 }

55 void LinkedList::insertNode(int data){

56 Node* newNode = new Node(data);

57 if (head == NULL) {

58 head = newNode;

59 return;

Ln 2, Col 14 Spaces: 4 UTF-8 CRLF C++ Win32

0 0

Type here to search

23°C Clear

ENG IN

1:07 AM

2/21/2023

File Edit Selection View Go Run ...

← →

Daa-code

screenrec

EXPLORER

DAA-CODE

> .vscode

> daacode-java

≡ a.exe

BF_string_matching.cpp

BinarySearch.cpp

BinarySearchRecursive.cpp

BubbleSort.cpp

exhaustive_search.cpp

HalfLinearSearch.cpp

InsertionSort.cpp

LinearSearch.cpp

LinearSearch.exe

LinkedList.cpp

MergeSort.cpp

QueueUsingArray.cpp

QuickSort.cpp

SelectionSort.cpp

StackUsingArray.cpp

StrassenseMatrixMultiplication....

travelling_salesman_problem.cpp

> OUTLINE

> TIMELINE

StackUsingArray.cpp

QueueUsingArray.cpp

LinkedList.cpp X

LinkedList.cpp > ...

```
55 void LinkedList::insertNode(int data){
56     Node* newNode = new Node(data);
57     if (head == NULL) {
58         head = newNode;
59         return;
60     }
61     Node* temp = head;
62     while (temp->next != NULL) {
63         temp = temp->next;
64     }
65     temp->next = newNode;
66 }
67 void LinkedList::printList(){
68     Node* temp = head;
69     if (head == NULL) {
70         cout << "List empty" << endl;
71         return;
72     }
73     while (temp != NULL) {
74         cout << temp->data << " ";
75         temp = temp->next;
76     }
77 }
78 int main(){
79     LinkedList list;
80     list.insertNode(1);
81     list.insertNode(2);
82     list.insertNode(3);
83     list.insertNode(4);
84     cout << "Elements of the list are: ";
85     list.printList();
86     cout << endl;
```

Ln 2, Col 14 Spaces: 4 UTF-8 CRLF C++ Win32

23°C Clear

ENG IN

1:07 AM

2/21/2023

File Edit Selection View Go Run ...

Daa-code

screenrec

EXPLORER

DAA-CODE

.vscode

daacode-java

a.exe

BF_string_matching.cpp

BinarySearch.cpp

BinarySearchRecursive.cpp

BubbleSort.cpp

exhaustive_search.cpp

HalfLinearSearch.cpp

InsertionSort.cpp

LinearSearch.cpp

LinearSearch.exe

LinkedList.cpp

MergeSort.cpp

QueueUsingArray.cpp

QuickSort.cpp

SelectionSort.cpp

StackUsingArray.cpp

StrassenseMatrixMultiplication....

travelling_salesman_problem.cpp

OUTLINE

TIMELINE

StackUsingArray.cpp

QueueUsingArray.cpp

LinkedList.cpp X

LinkedList.cpp > ...

```
76
77 }
78 int main(){
79     LinkedList list;
80     list.insertNode(1);
81     list.insertNode(2);
82     list.insertNode(3);
83     list.insertNode(4);
84     cout << "Elements of the list are: ";
85     list.printList();
86     cout << endl;
87     list.deleteNode(2);
88     cout << "Elements of the list are: ";
89     list.printList();
90     cout << endl;
91     return 0;
92 }
93
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

```
PS D:\DAA\Daa-code> g++ LinkedList.cpp
PS D:\DAA\Daa-code> .\a.exe
Elements of the list are: 1 2 3 4
Elements of the list are: 1 3 4
PS D:\DAA\Daa-code> 
```

+ ... ^ x

powershell

powershell

Ln 2, Col 14 Spaces: 4 UTF-8 CRLF C++ Win32

22°C Clear 1:08 AM 2/21/2023

File Edit Selection View Go Run ... Daa-code

EXPLORER

- ▼ DAA-CODE
 - > .vscode
 - > daacode-java
 - a.exe
 - BF_string_matching.cpp
 - BinarySearch.cpp
 - BinarySearchRecursive.cpp
 - BubbleSort.cpp
 - exhaustive_search.cpp
 - HalfLinearSearch.cpp
 - InsertionSort.cpp
 - LinearSearch.cpp
 - LinearSearch.exe
 - LinkedList.cpp
 - MergeSort.cpp
 - QueueUsingArray.cpp
 - QuickSort.cpp
 - SelectionSort.cpp
 - StackUsingArray.cpp
 - StrassenseMatrixMultiplication....
 - travelling_salesman_problem.cpp
 - Tree.cpp
- OUTLINE
- TIMELINE

StackUsingArray.cpp QueueUsingArray.cpp LinkedList.cpp Tree.cpp

Tree.cpp > ...

```
1 //Name:Maniyar Danish, Reg.no:2020BIT028
2 //Tree
3 #include <iostream>
4 struct Node {
5     int data;
6     Node* left;
7     Node* right;
8     Node(int data) {
9         this->data = data;
10        this->left = nullptr;
11        this->right = nullptr;
12    }
13 };
14 class BinaryTree {
15 public:
16     Node* root;
17     BinaryTree() {
18         root = nullptr;
19     }
20     void addNode(int data) {
21         Node* newNode = new Node(data);
22         if (root == nullptr) {
23             root = newNode;
24         } else {
25             Node* focusNode = root;
26             Node* parent;
27             while (true) {
28                 parent = focusNode;
29                 if (data < focusNode->data) {
30                     focusNode = focusNode->left;
31                 }
32                 if (focusNode == nullptr) {
33                     parent->left = newNode;
```

Ln 2, Col 7 Spaces: 4 UTF-8 CRLF C++ Win32

Type here to search

22°C Clear ENG IN 1:33 AM 2/21/2023

FileEditSelectionViewGoRun...Daa-code

stackUsingArray.cppQueueUsingArray.cppLinkedList.cppTree.cpp

screenrec

EXPLORER

DAA-CODE

.vscode

daacode-java

a.exe

BF_string_matching.cpp

BinarySearch.cpp

BinarySearchRecursive.cpp

BubbleSort.cpp

exhaustive_search.cpp

HalfLinearSearch.cpp

InsertionSort.cpp

LinearSearch.cpp

LinearSearch.exe

LinkedList.cpp

MergeSort.cpp

QueueUsingArray.cpp

QuickSort.cpp

SelectionSort.cpp

StackUsingArray.cpp

StrassenseMatrixMultiplication....

travelling_salesman_problem.cpp

Tree.cpp

OUTLINE

TIMELINE

Tree.cpp > ...

```
28     parent = focusNode;
29     if (data < focusNode->data) {
30         focusNode = focusNode->left;
31         if (focusNode == nullptr) {
32             parent->left = newNode;
33             return;
34         }
35     } else {
36         focusNode = focusNode->right;
37         if (focusNode == nullptr) {
38             parent->right = newNode;
39             return;
40         }
41     }
42 }
43 }
44 }
45 void preOrderTraversal(Node* focusNode) {
46     if (focusNode != nullptr) {
47         std::cout << focusNode->data << " ";
48         preOrderTraversal(focusNode->left);
49         preOrderTraversal(focusNode->right);
50     }
51 }
52 };
53 int main() {
54     BinaryTree tree;
55     tree.addNode(50);
56     tree.addNode(25);
57     tree.addNode(75);
58     tree.addNode(12);
59     tree.addNode(37);
```

Ln 2, Col 7 Spaces: 4 UTF-8 CRLF C++ Win32

Type here to search

22°C Clear

ENG IN 1:33 AM 2/21/2023

File Edit Selection View Go Run ...

← →

Daa-code

screenrec

EXPLORER

DAA-CODE

> .vscode

> daacode-java

≡ a.exe

BF_string_matching.cpp

BinarySearch.cpp

BinarySearchRecursive.cpp

BubbleSort.cpp

exhaustive_search.cpp

HalfLinearSearch.cpp

InsertionSort.cpp

LinearSearch.cpp

LinearSearch.exe

LinkedList.cpp

MergeSort.cpp

QueueUsingArray.cpp

QuickSort.cpp

SelectionSort.cpp

StackUsingArray.cpp

StrassenseMatrixMultiplication....

travelling_salesman_problem.cpp

Tree.cpp

OUTLINE

TIMELINE

StackUsingArray.cpp

QueueUsingArray.cpp

LinkedList.cpp

Tree.cpp

Tree.cpp > ...

51 }

52 };

53 int main() {

54 BinaryTree tree;

55 tree.addNode(50);

56 tree.addNode(25);

57 tree.addNode(75);

58 tree.addNode(12);

59 tree.addNode(37);

60 tree.addNode(43);

61 tree.addNode(30);

62 tree.preOrderTraversal(tree.root);

63 return 0;

64 }

65 |

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

50 25 12 37 30 43 75

PS D:\DAA\Daa-code>

+ ... ^ x

powershell

powershell

Ln 65, Col 1 Spaces: 4 UTF-8 CRLF C++ Win32

Type here to search

22°C Clear

ENG IN

1:33 AM

2/21/2023

File Edit Selection View Go Run ...

Daa-code

screenrec

EXPLORER

DAA-CODE

.vscode

daacode-java

a.exe

BF_string_matching.cpp

BinarySearch.cpp

BinarySearchRecursive.cpp

BubbleSort.cpp

exhaustive_search.cpp

Graph.cpp

HalfLinearSearch.cpp

InsertionSort.cpp

LinearSearch.cpp

LinearSearch.exe

LinkedList.cpp

MergeSort.cpp

QueueUsingArray.cpp

QuickSort.cpp

SelectionSort.cpp

StackUsingArray.cpp

StrassenseMatrixMultiplication....

travelling_salesman_problem.cpp

Tree.cpp

Graph.cpp > ...

```
1 //Name: Maniyar Danish, Reg.no:2020BIT028
2 // Graph
3 #include <bits/stdc++.h>
4 using namespace std;
5 void addEdge(vector<int> adj[], int u, int v){
6     adj[u].push_back(v);
7     adj[v].push_back(u);
8 }
9 void printGraph(vector<int> adj[], int V){
10     for (int v = 0; v < V; ++v) {
11         cout << "\n Adjacency list of vertex " << v
12             << "\n head ";
13         for (auto x : adj[v])
14             cout << "-> " << x;
15         cout<<endl;
16     }
17 }
18 int main(){
19     int V = 5;
20     vector<int> adj[V];
21     addEdge(adj, 0, 1);
22     addEdge(adj, 0, 4);
23     addEdge(adj, 1, 2);
24     addEdge(adj, 1, 3);
25     addEdge(adj, 1, 4);
26     addEdge(adj, 2, 3);
27     addEdge(adj, 3, 4);
28     printGraph(adj, V);
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

Adjacency list of vertex 1
head -> 0-> 2-> 3-> 4

Ln 2, Col 9 Spaces: 4 UTF-8 CRLF C++ Win32

22°C Clear

ENG IN

1:37 AM

2/21/2023

