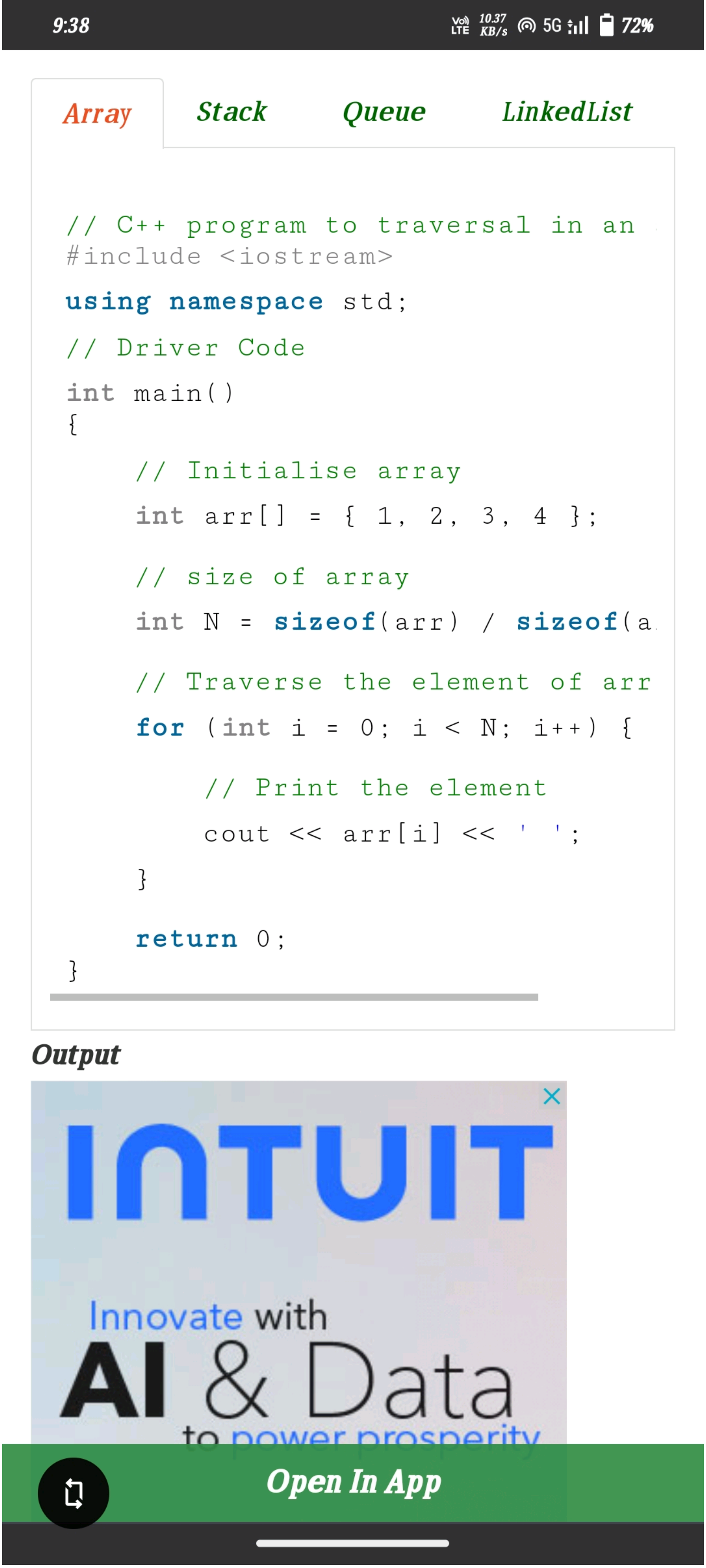
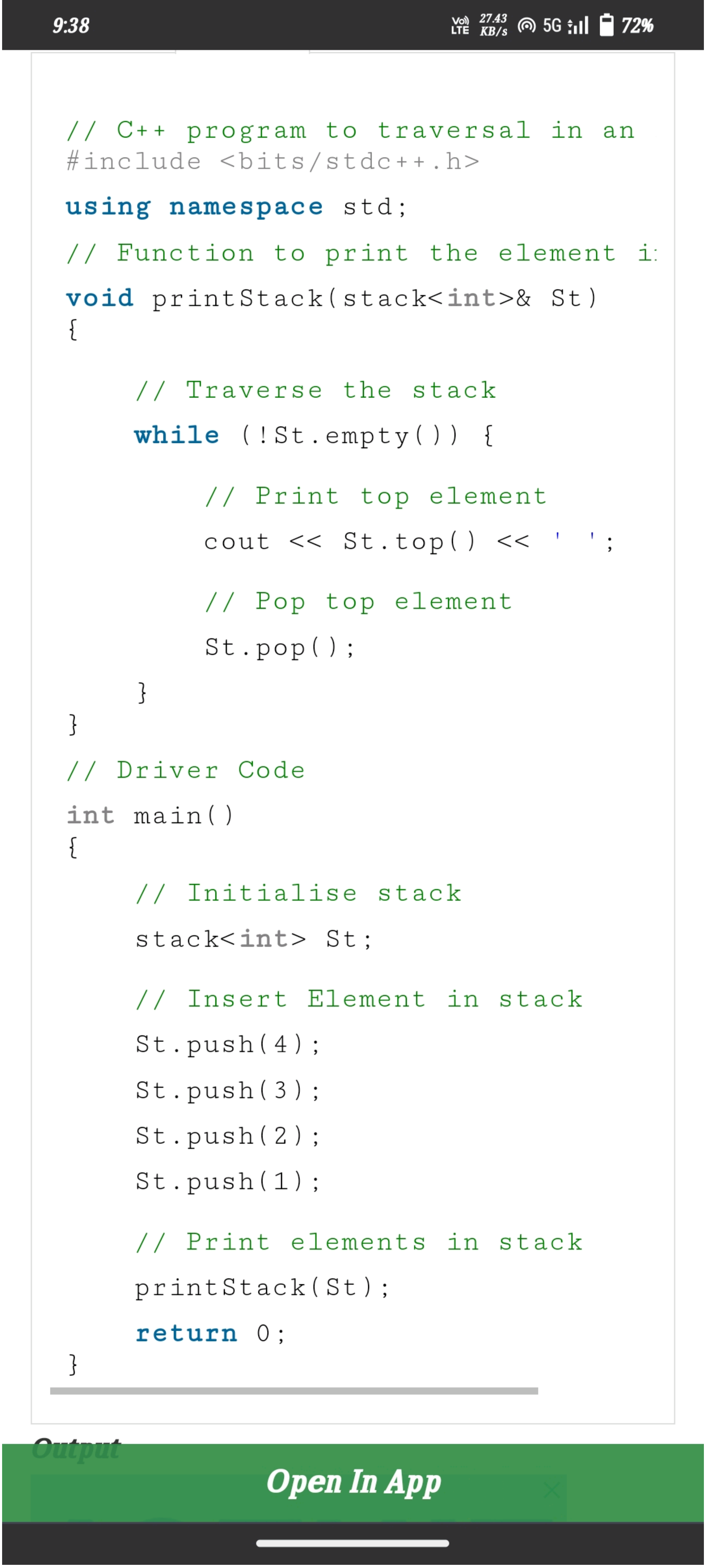


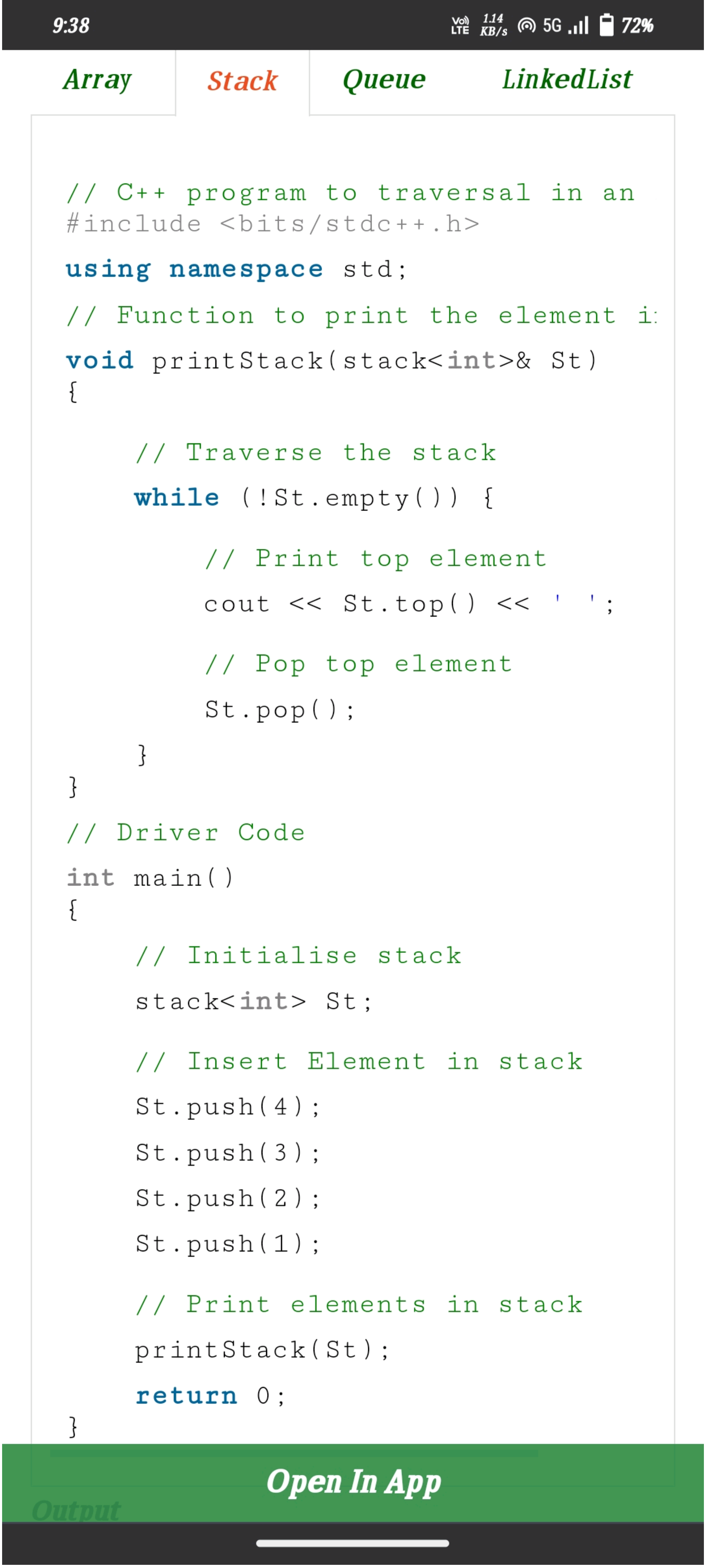
```
// C++ program to traversal in an
#include <iostream>
using namespace std;
// Driver Code
int main()
{
    // Initialise array
    int arr[] = { 1, 2, 3, 4 };

    // size of array
    int N = sizeof(arr) / sizeof(a

    // Traverse the element of arr
    for (int i = 0; i < N; i++) {

        // Print the element
        cout << arr[i] << ' ';
    }

    return 0;
}
```

Output

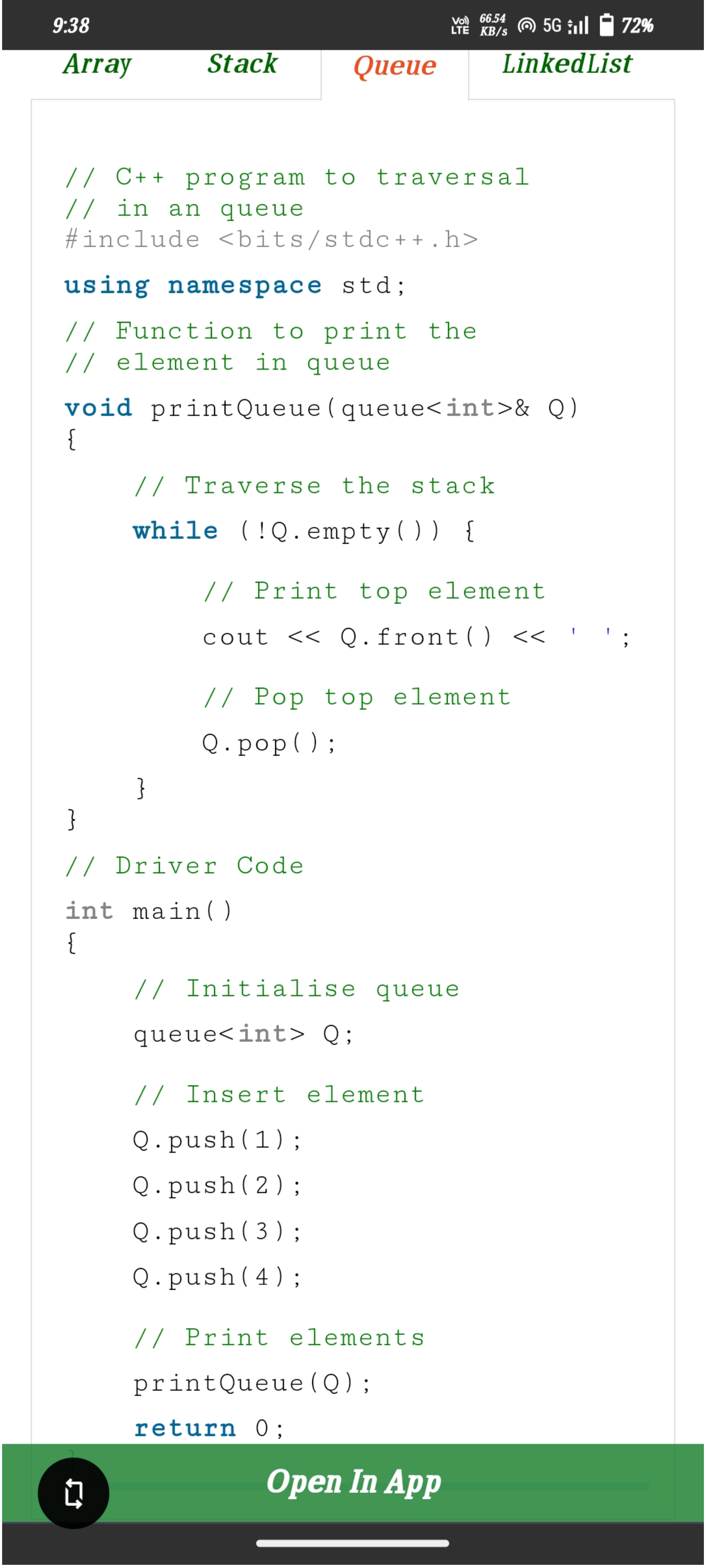


```
// C++ program to traversal in an
#include <bits/stdc++.h>
using namespace std;

// Function to print the element i
void printStack(stack<int>& St)
{

    // Traverse the stack
    while (!St.empty()) {

        // Print top element
        cout << St.top() << ' ';

        // Pop top element
        St.pop();
    }
}

// Driver Code
int main()
{

    // Initialise stack
    stack<int> St;

    // Insert Element in stack
    St.push(4);
    St.push(3);
    St.push(2);
    St.push(1);

    // Print elements in stack
    printStack(St);

    return 0;
}
```

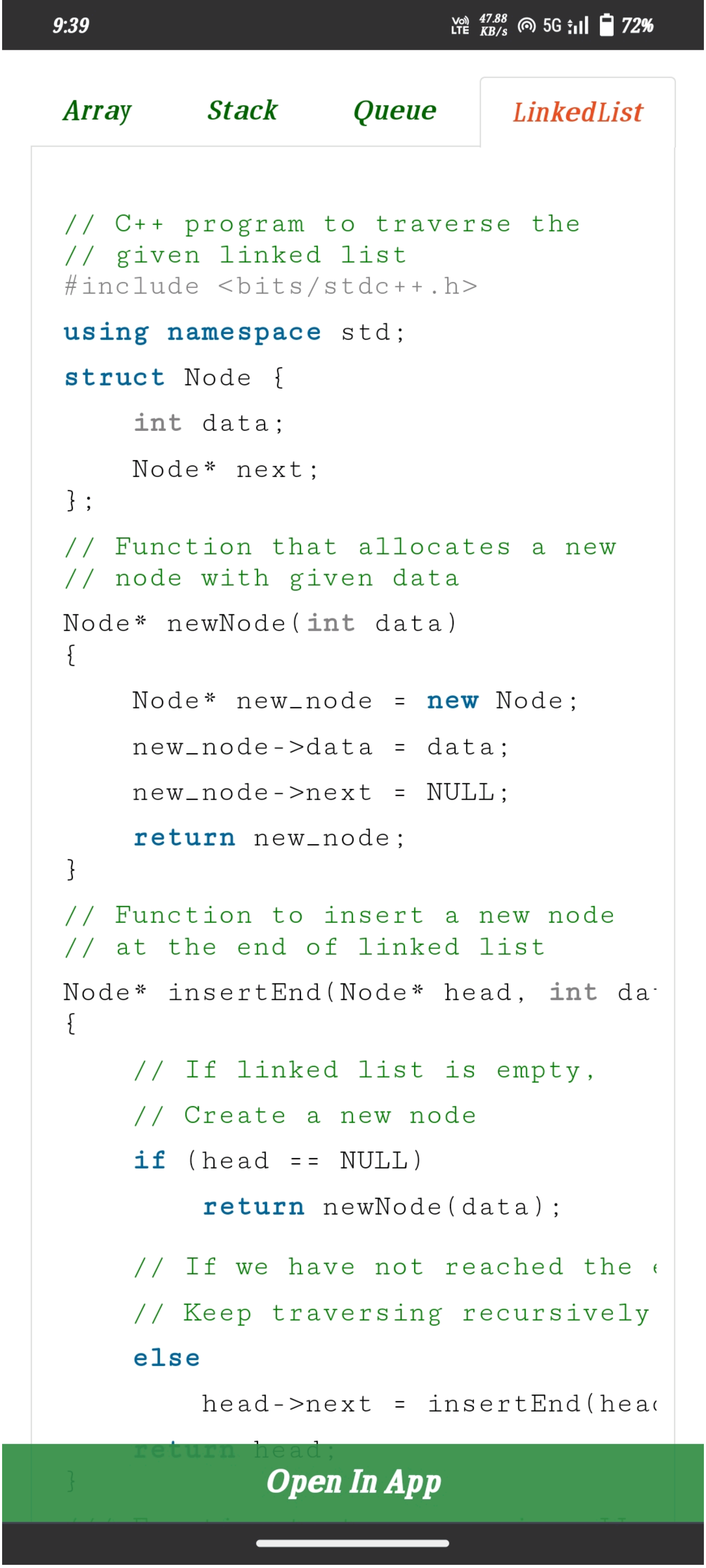
Output



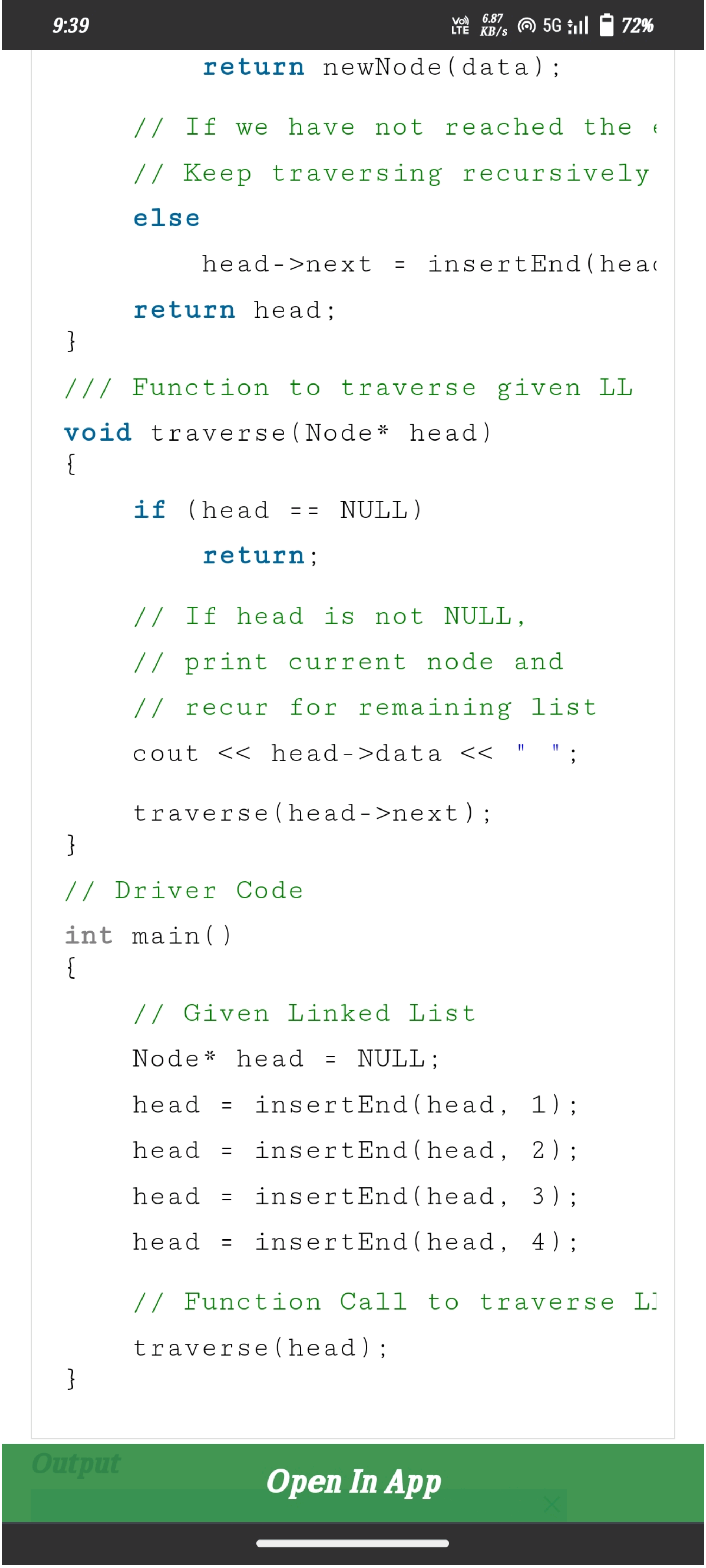
```
// C++ program to traversal in an
#include <bits/stdc++.h>
using namespace std;
// Function to print the element i
void printStack(stack<int>& St)
{
    // Traverse the stack
    while (!St.empty()) {
        // Print top element
        cout << St.top() << ' ';
        // Pop top element
        St.pop();
    }
}
// Driver Code
int main()
{
    // Initialise stack
    stack<int> St;
    // Insert Element in stack
    St.push(4);
    St.push(3);
    St.push(2);
    St.push(1);
    // Print elements in stack
    printStack(St);
    return 0;
}
```



```
// C++ program to traversal
// in an queue
#include <bits/stdc++.h>

using namespace std;

// Function to print the
// element in queue

void printQueue(queue<int>& Q)
{

    // Traverse the stack

    while (!Q.empty()) {

        // Print top element

        cout << Q.front() << ' ';

        // Pop top element

        Q.pop();

    }
}

// Driver Code

int main()
{

    // Initialise queue

    queue<int> Q;

    // Insert element

    Q.push(1);

    Q.push(2);

    Q.push(3);

    Q.push(4);

    // Print elements

    printQueue(Q);

    return 0;
}
```



```
// C++ program to traverse the
// given linked list
#include <bits/stdc++.h>
using namespace std;
struct Node {
    int data;
    Node* next;
};
// Function that allocates a new
// node with given data
Node* newNode(int data)
{
    Node* new_node = new Node;
    new_node->data = data;
    new_node->next = NULL;
    return new_node;
}
// Function to insert a new node
// at the end of linked list
Node* insertEnd(Node* head, int da
{
    // If linked list is empty,
    // Create a new node
    if (head == NULL)
        return newNode(data);
    // If we have not reached the e
    // Keep traversing recursively
    else
        head->next = insertEnd(hea
    return head;
}
```



```
        return newNode(data);

    // If we have not reached the
    // Keep traversing recursively
    else
        head->next = insertEnd(hea

    return head;
}
/// Function to traverse given LL
void traverse(Node* head)
{
    if (head == NULL)
        return;

    // If head is not NULL,
    // print current node and
    // recur for remaining list
    cout << head->data << " ";

    traverse(head->next);
}
// Driver Code
int main()
{
    // Given Linked List
    Node* head = NULL;
    head = insertEnd(head, 1);
    head = insertEnd(head, 2);
    head = insertEnd(head, 3);
    head = insertEnd(head, 4);

    // Function Call to traverse LI
    traverse(head);
}
```