

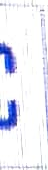
```
2  #include <bits/stdc++.h>
3  using namespace std;
4  class Graph {
5  public:
6      map<int, bool> visited;
7      map<int, list<int> > adj;
8      void addEdge(int v, int w);
9      void DFS(int v);
10 };
11
12 void Graph::addEdge(int v, int w)
13 {
14     adj[v].push_back(w);
15 }
16
17 void Graph::DFS(int v)
18 {
19     visited[v] = true;
20     cout << v << " ";
21     list<int>::iterator i;
22     for (i = adj[v].begin(); i != adj[v].end(); ++i)
23         if (!visited[*i])
24             DFS(*i);
25 }
26
27 int main()
28 {
29     Graph g;
30     g.addEdge(0, 1);
31     g.addEdge(0, 2);
32     g.addEdge(1, 2);
33     g.addEdge(2, 0);
34     g.addEdge(2, 3);
35     g.addEdge(3, 3);
36     cout << "Following is Depth First Traversal"
37          << " (starting from vertex 2) \n";
38     g.DFS(2);
39     return 0;
```


main.cpp

```
1  #include<iostream>
2  #include <queue>
3
4  using namespace std;
5
6  int main() {
7      queue<string> animals;
8
9      // push elements into the queue
10     animals.push("Cat");
11     animals.push("Dog");
12
13     cout << "Queue: ";
14     while(!animals.empty()) {
15         cout << animals.front() << ", ";
16         animals.pop();
17     }
18
19     cout << endl;
20
21     return 0;
22 }
```


main.cpp

```
1  include <iostream>
2  using namespace std;
3  struct Node {
4      int data;
5      struct Node *next;
6  };
7  struct Node* head = NULL;
8  void insert(int new_data) {
9      struct Node* new_node = (struct Node*) malloc(sizeof(struct Node));
10     new_node->data = new_data;
11     new_node->next = head;
12     head = new_node;
13 }
14 void display() {
15     struct Node* ptr;
16     ptr = head;
17     while (ptr != NULL) {
18         cout<< ptr->data <<" ";
19         ptr = ptr->next;
20     }
21 }
22 int main() {
23     insert(3);
24     insert(1);
25     insert(7);
26     insert(2);
27     insert(9);
28     cout<<"The linked list is: ";
29     display();
30     return 0;
31 }
32
```

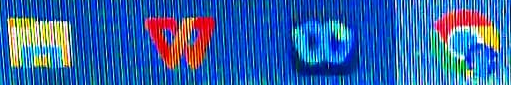
Run

Output

/tmp/3XDNKfT6Ks.o

Python

I




```
2  #include <bits/stdc++.h>
3  using namespace std;
4  struct Node {
5      int data;
6      struct Node *left, *right;
7  };
8  Node* newNode(int data)
9  {
10     Node* temp = new Node;
11     temp->data = data;
12     temp->left = temp->right = NULL;
13     return temp;
14 }
15
16 void printInorder(struct Node* node)
17 {
18     if (node == NULL)
19         return;
20     printInorder(node->left);
21     cout << node->data << " ";
22     printInorder(node->right);
23 }
24 int main()
25 {
26     struct Node* root = newNode(1);
27     root->left = newNode(2);
28     root->right = newNode(3);
29     root->left->left = newNode(4);
30     root->left->right = newNode(5);
31     cout << "\nInorder traversal of binary tree is \n";
32     printInorder(root);
33     return 0;
34 }
```