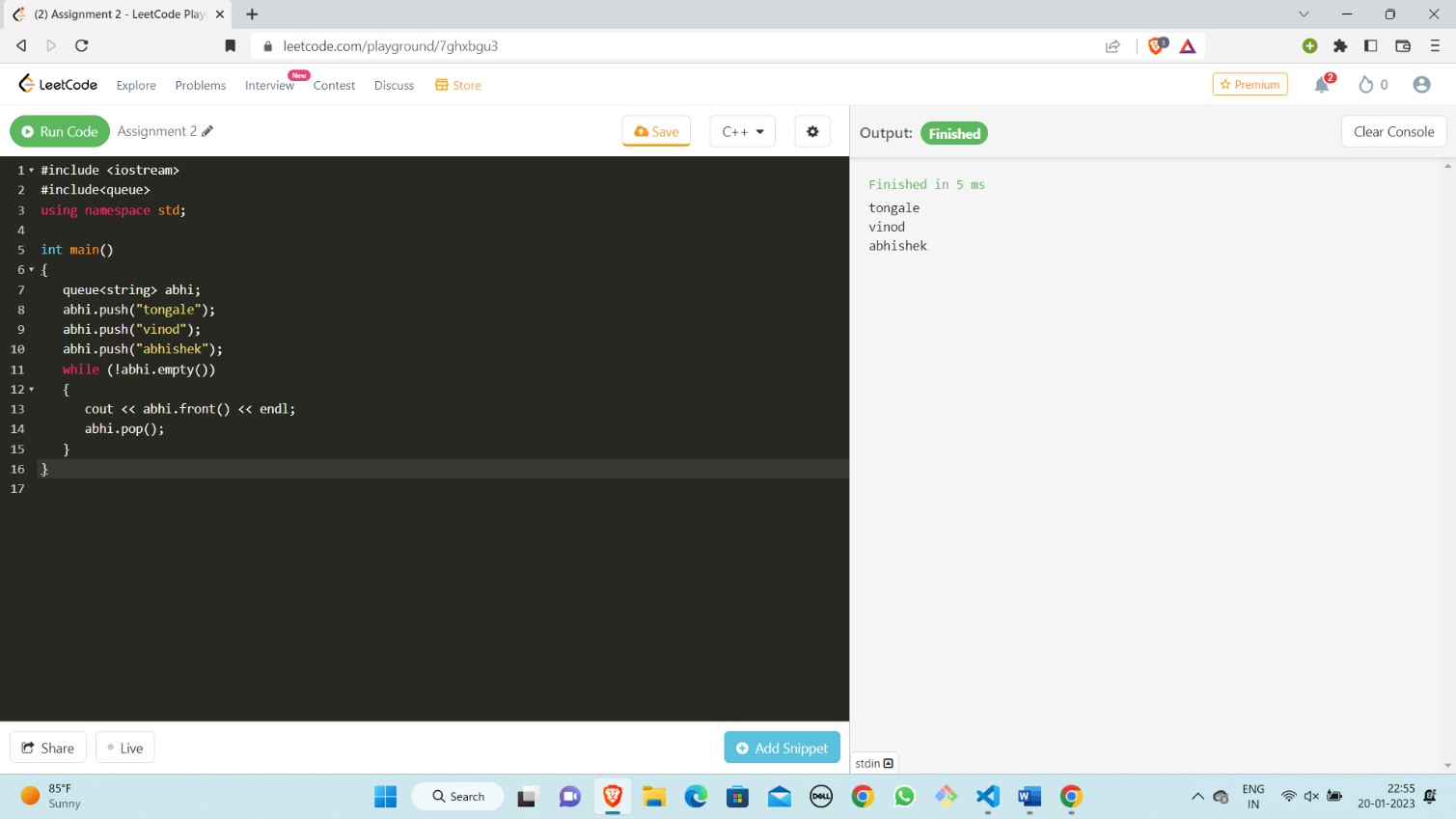
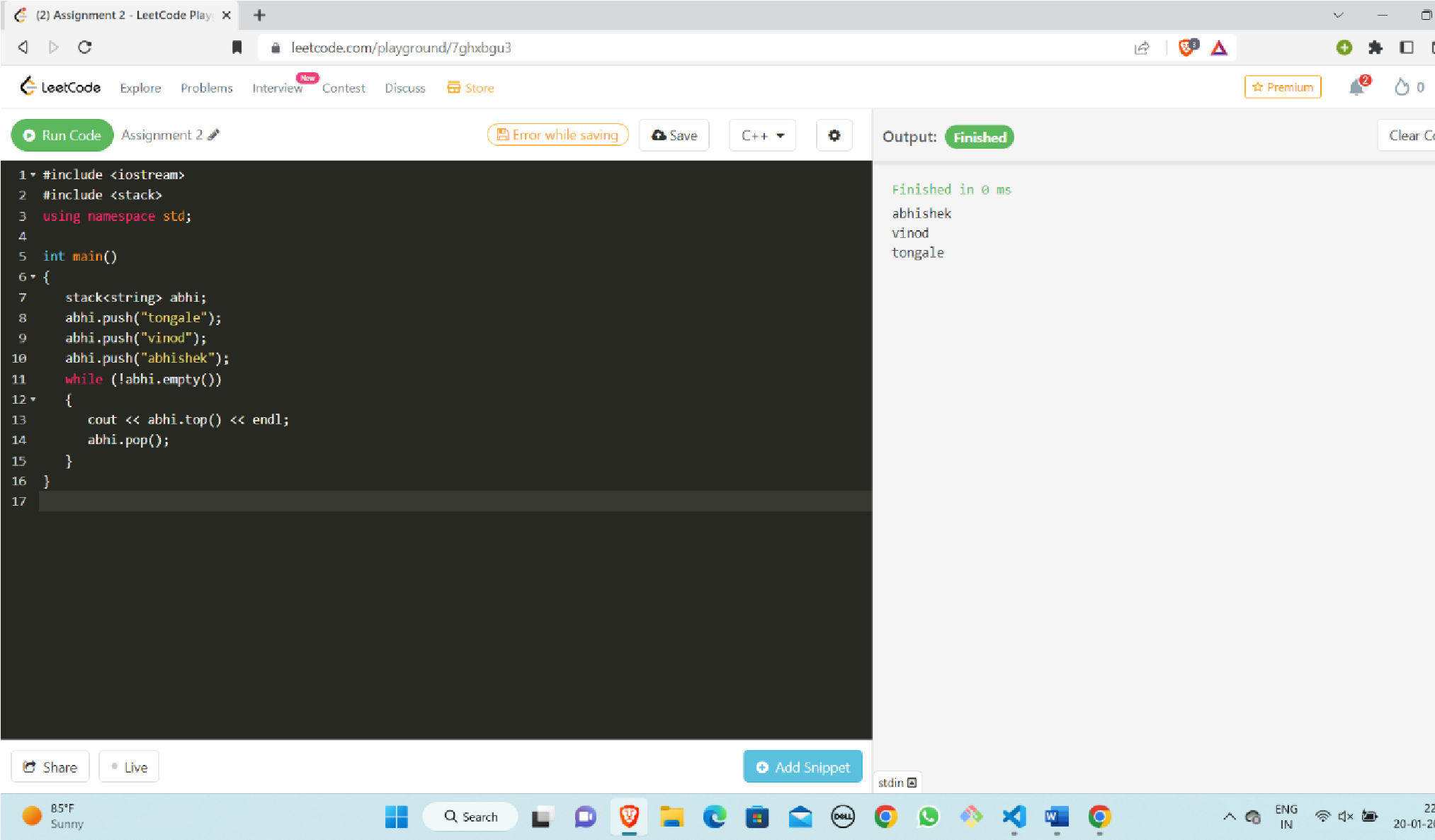
**Assignment - 1**

Name : Abhishek satapure

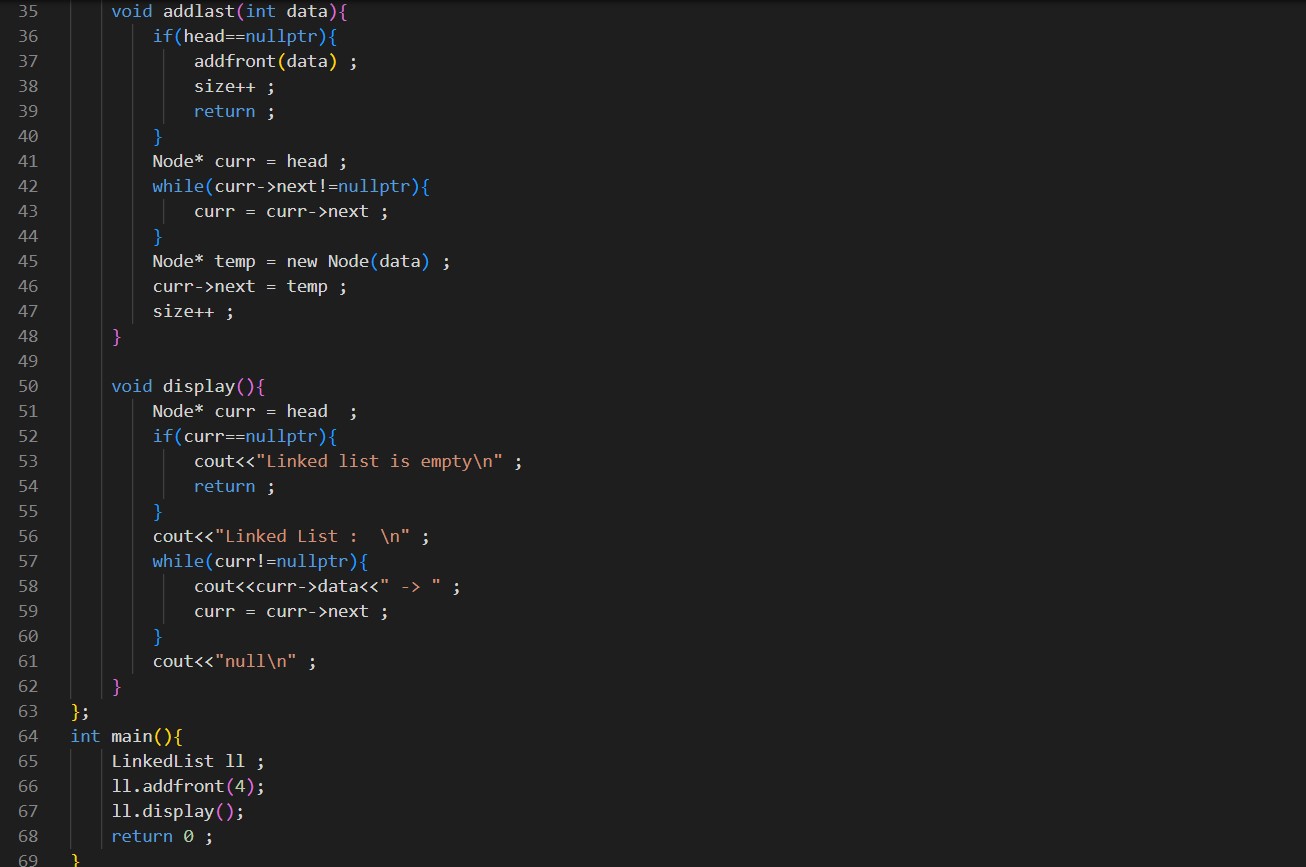
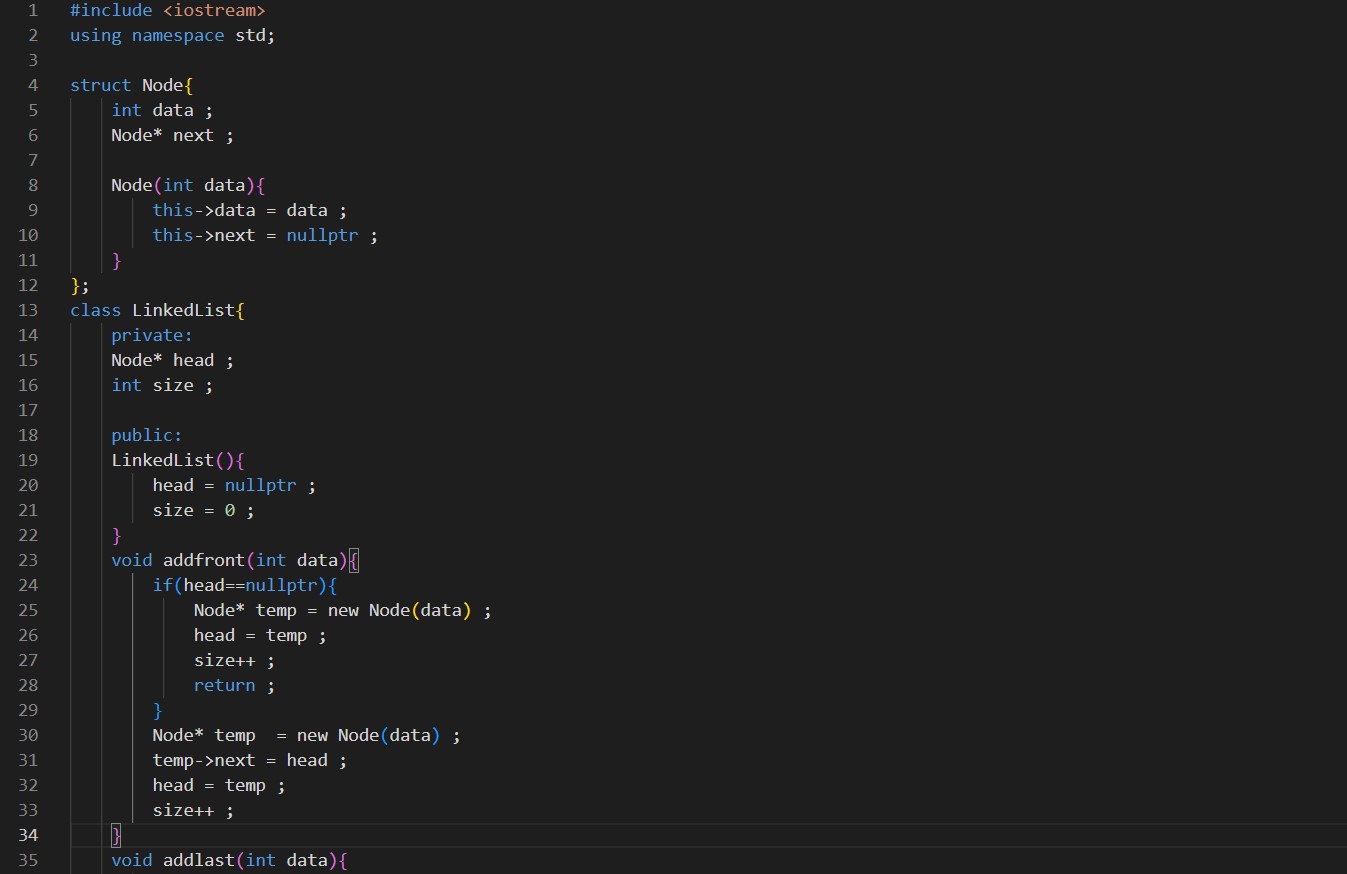
1. Queue



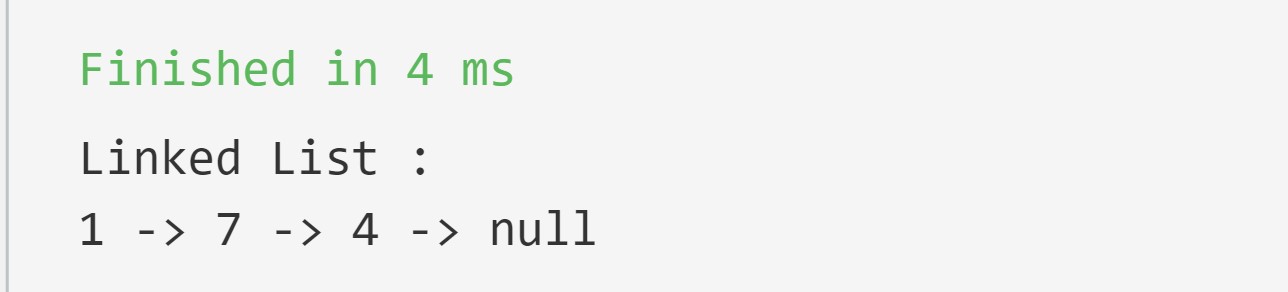
1. **Stack**

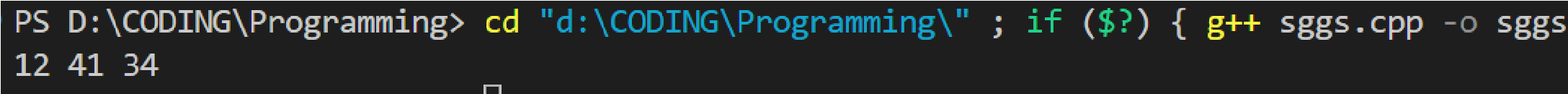


1. **Linked List:**



**Output:**



1. **Tree :**

|  |
| --- |
| #include<iostream>  #include<cstdlib> using namespace std; struct node{ int data; struct node \*left; struct node \*right;  };  void preOrder(struct node\*root){ if(root!=NULL){ cout<<root->data<<" "; preOrder(root->left); preOrder(root->right);  }  }  struct node\* createNode(int data){ struct node \*ptr=(struct node\*)malloc(sizeof(struct node)); ptr->data=data; ptr->left=NULL; ptr->right=NULL; return ptr;  } int main(){ struct node\* p=createNode(12); struct node\*p1=createNode(41); struct node\*p2=createNode(34); p->left=p1; p->right=p2; preOrder(p); return 0;} |

**Output:**

1. **Graph:**

|  |
| --- |
| #include <iostream>  #include <vector> using namespace std;  struct Edge { int src, dest;  };  class Graph  { public:  vector<vector<int>> adjList;    Graph(vector<Edge> const &edges, int n)  { adjList.resize(n);  for (auto &edge: edges)  { adjList[edge.src].push\_back(edge.dest);    }  }  }; void printGraph(Graph const &graph, int n)  { for (int i = 0; i < n; i++)  { cout << i << "-";  for (int v: graph.adjList[i]) { cout << v << " ";  } cout << endl;  }  }  int main()  { vector<Edge> edges =  {  {0, 1}, {1, 2}, {2, 0}, {2, 1}, {3, 2}, {4, 5}, {5, 4}  }; int n = 6;    Graph graph(edges, n); |
| printGraph(graph, n);  return 0;  } |



**OutPut:**