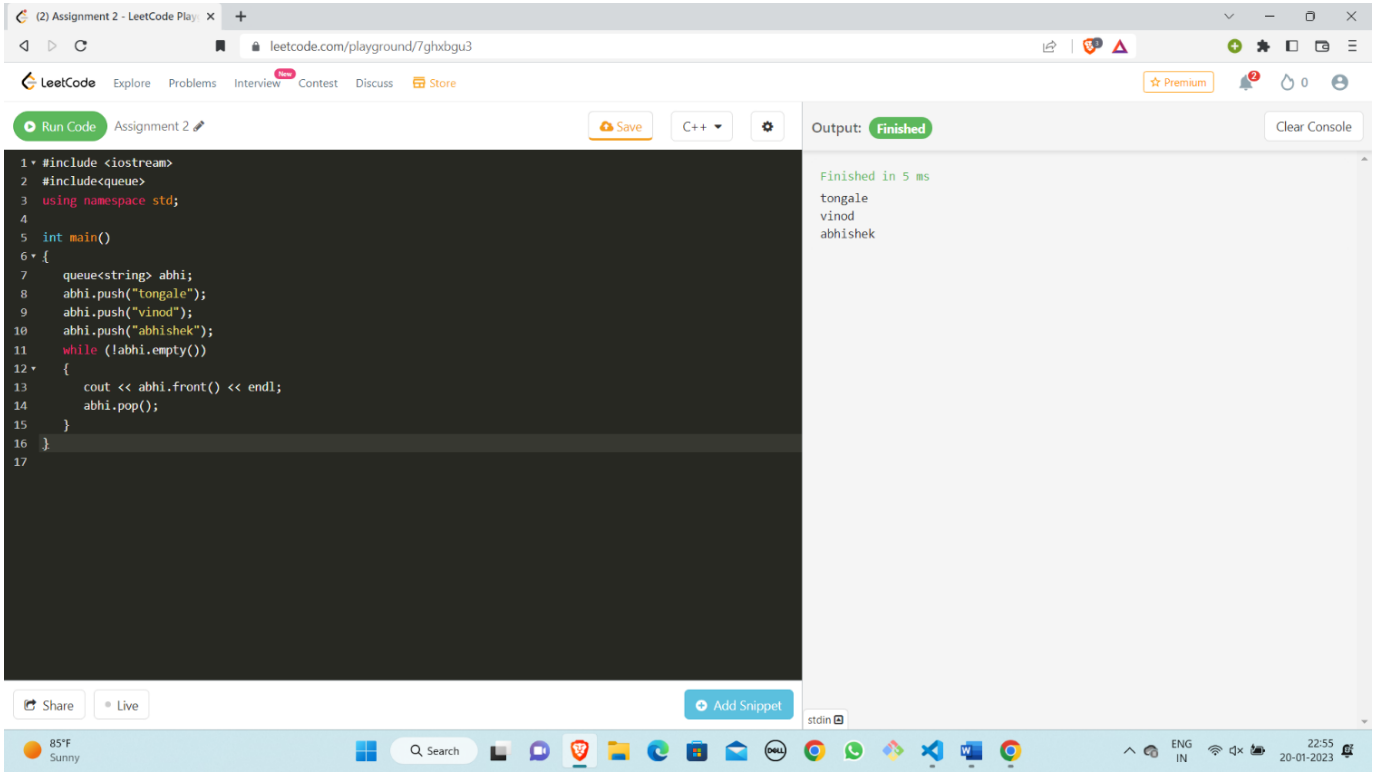
Practical-1

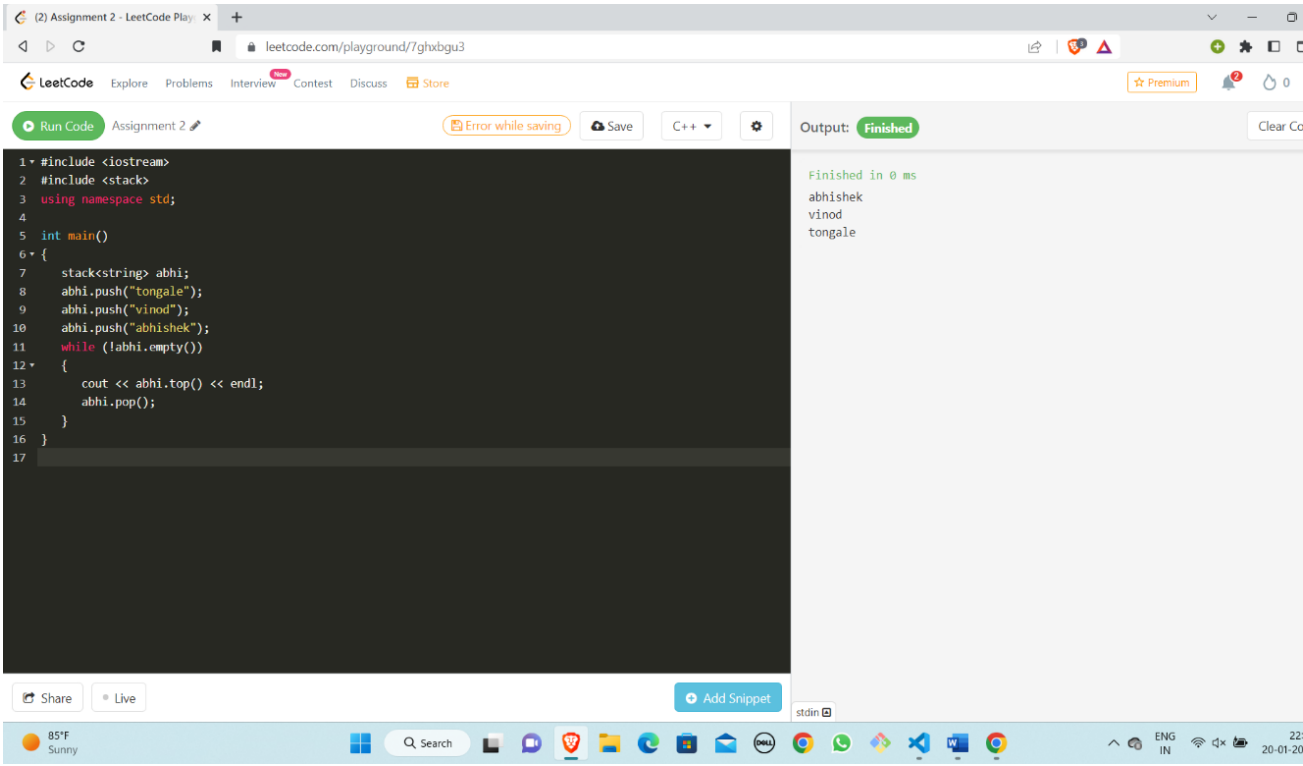
Name : sanket Gaikwad

Reg.No : 2020BIT036

1) Queue



2) stack


(2) Assignment 2 - LeetCode Play
leetcode.com/playground/7ghxbgu3
LeetCode
Explore
Problems
Interview
Contest
Discuss
Store
Premium
Run Code
Assignment 2
Error while saving
Save
C++
Output:
Finished
Clear Co
#include <iostream>
#include <stack>
using namespace std;
int main()
{
stack<string> abhi;
abhi.push("tongale");
abhi.push("vinod");
abhi.push("abhishek");
while (!abhi.empty())
{
cout << abhi.top() << endl;
abhi.pop();
}
}
Finished in 0 ms
abhishek
vinod
tongale
Share
Live
Add Snippet
stdin
85°F
Sunny
Search
ENG
IN
20-01-20

3) Linked List:

```cpp
#include <iostream>
using namespace std;

struct Node{
    int data ;
    Node* next ;

    Node(int data){
        this->data = data ;
        this->next = nullptr ;
    }
};
class LinkedList{
    private:
    Node* head ;
    int size ;

    public:
    LinkedList(){
        head = nullptr ;
        size = 0 ;
    }
    void addfront(int data){
        if(head==nullptr){
            Node* temp = new Node(data) ;
            head = temp ;
            size++ ;
            return ;
        }
        Node* temp  = new Node(data) ;
        temp->next = head ;
        head = temp ;
        size++ ;
    }
    void addlast(int data){
```

```cpp
    void addlast(int data){
        if(head==nullptr){
            addfront(data) ;
            size++ ;
            return ;
        }
        Node* curr = head ;
        while(curr->next!=nullptr){
            curr = curr->next ;
        }
        Node* temp = new Node(data) ;
        curr->next = temp ;
        size++ ;
    }

    void display(){
        Node* curr = head  ;
        if(curr==nullptr){
            cout<<"Linked list is empty\n" ;
            return ;
        }
        cout<<"Linked List :  \n" ;
        while(curr!=nullptr){
            cout<<curr->data<<" -> " ;
            curr = curr->next ;
        }
        cout<<"null\n" ;
    }
};
int main(){
    LinkedList ll ;
    ll.addfront(4);
    ll.display();
    return 0 ;
}
```

Output:

```
Finished in 4 ms
Linked List :
1 -> 7 -> 4 -> null
```

4) 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:

```
PS D:\CODING\Programming> cd "d:\CODING\Programming\" ; if ($?) { g++ sggs.cpp -o sggs
12 41 34
```

5) 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:

```
0-1
1-2
2-0 1
3-2
4-5
5-4
PS D:\CODING\Programming>
```