Q1Solution:-

const express=require('express');

const bodyParser=require('body-parser')

const app=express();

app.use(express.urlencoded());

const sayHi=(req,res)=>{

return res.json(200,{

message:"Hi buddy it's working fine!"

});

}

app.get('/',sayHi);

app.post('/ChemicalsName',(req,res)=>{

res.send(req.body);

});

app.listen(8000,()=>{

console.log('Server is running on port 8000');

})

Q2Solution:-

#include<bits/stdc++.h>

using namespace std;

void Intersection(int num1[],int n1,int num2[],int n2)

{

map<int,int> m1;

map<int,int> m2;

vector<int> v;

for(int i=0;i<n1;i++)

{

m1[num1[i]]++;

}

for(int i=0;i<n2;i++)

{

m2[num2[i]]++;

}

for(auto it=m1.begin();it!=m1.end();it++)

{

if(m2.find(it->first)!=m2.end())

{

v.push\_back(it->first);

}

}

for(int i=0;i<v.size();i++)

{

cout<<v[i]<<" ";

}

}

int main()

{

int n1,n2;

cin>>n1>>n2;

int num1[n1];

int num2[n2];

for(int i=0;i<n1;i++)

{

cin>>num1[i];

}

for(int i=0;i<n2;i++)

{

cin>>num2[i];

}

Intersection(num1,n1,num2,n2);

}

Q3Solution:-

#include<bits/stdc++.h>

using namespace std;

bool checkForValid(string str)

{

stack<char> s;

for(int i=0;i<str.length();i++)

{

if(str[i]=='{'|| str[i]=='['|| str[i]=='(')

{

s.push(str[i]);

}

else

{

if(str[i]=='}' && s.top()=='{')

{

s.pop();

}

else if(str[i]==']' && s.top()=='[')

{

s.pop();

}

else if(str[i]==')' && s.top()=='(')

{

s.pop();

}

else{

s.push(str[i]);

}

}

}

if(s.empty())

{

return true;

}

else{

return false;

}

}

int main()

{

string str;

cin>>str;

if(checkForValid(str))

{

cout<<"true";

}

else

{

cout<<"false";

}

}

Q4Solution:-

#include<bits/stdc++.h>

using namespace std;

int main()

{

map<int,int> m;

int n1,n2;

cin>>n1;

int num1[n1];

for(int i=0;i<n1;i++)

{

cin>>num1[i];

}

for(int i=0;i<n1;i++)

{

m[num1[i]]++;

}

for(auto it=m.begin();it!=m.end();it++)

{

if(it->second==1)

{

cout<<it->first;

}

}

}