**Question 2.**

# Python program to illustrate the intersection

# of two lists in most simple way

def intersection(lst1, lst2):

lst3 = [value for value in lst1 if value in lst2]

return lst3

# Driver Code

lst1 = [4, 9, 1, 17, 11, 26, 28, 54, 69]

lst2 = [9, 9, 74, 21, 45, 11, 63, 28, 26]

print(intersection(lst1, lst2)

**Question 3**.

open\_list = ["[","{","("]

close\_list = ["]","}",")"]

# Function to check parentheses

def check(myStr):

stack = []

for i in myStr:

if i in open\_list:

stack.append(i)

elif i in close\_list:

pos = close\_list.index(i)

if ((len(stack) > 0) and

(open\_list[pos] == stack[len(stack)-1])):

stack.pop()

else:

return "Unbalanced"

if len(stack) == 0:

return "Balanced"

else:

return "Unbalanced"

# Driver code

string = "{[]{()}}"

print(string,"-", check(string))

string = "[{}{})(]"

print(string,"-", check(string))

string = "((()"

print(string,"-",check(string))

**Question 4:**

# Python program to check if two

# to get unique values from list

# using traversal

# function to get unique values

def unique(list1):

# intilize a null list

unique\_list = []

# traverse for all elements

for x in list1:

# check if exists in unique\_list or not

if x not in unique\_list:

unique\_list.append(x)

# print list

for x in unique\_list:

print x,

# driver code

list1 = [10, 20, 10, 30, 40, 40]

print("the unique values from 1st list is")

unique(list1)

list2 =[1, 2, 1, 1, 3, 4, 3, 3, 5]

print("\nthe unique values from 2nd list is")

unique(list2)

**Quedtion 1:**:

const userSchema = new Schema({

firstName: String,

lastName: String,

email: String,

password: String,

permissionLevel: Number

});

const userModel = mongoose.model('Users', userSchema);

app.post('/users', [

UsersController.insert

]);

exports.insert = (req, res) => {

let salt = crypto.randomBytes(16).toString('base64');

let hash = crypto.createHmac('sha512',salt)

.update(req.body.password)

.digest("base64");

req.body.password = salt + "$" + hash;

req.body.permissionLevel = 1;

UserModel.createUser(req.body)

.then((result) => {

res.status(201).send({id: result.\_id});

});

};

{

"firstName" : "Marcos",

"lastName" : "Silva",

"email" : "marcos.henrique@toptal.com",

"password" : "s3cr3tp4sswo4rd"

}

fetch('http://localhost:3600/users', {

method: 'POST',

headers: {

"Content-type": "application/json"

},

body: JSON.stringify({

"firstName": "Marcos",

"lastName": "Silva",

"email": "marcos.henrique@toptal.com",

"password": "s3cr3tp4sswo4rd"

})

})

.then(function(response) {

return response.json();

})

.then(function(data) {

console.log('Request succeeded with JSON response', data);

})

.catch(function(error) {

console.log('Request failed', error);

});

exports.createUser = (userData) => {

const user = new User(userData);

return user.save();

};

/users/routes/config.js:

app.get('/users/:userId', [

UsersController.getById

])

/users/controllers/users.controller.js:

exports.getById = (req, res) => {

UserModel.findById(req.params.userId).then((result) => {

res.status(200).send(result);

});

};

exports.findById = (id) => {

return User.findById(id).then((result) => {

result = result.toJSON();

delete result.\_id;

delete result.\_\_v;

return result;

});

};

{

"firstName": "Marcos",

"lastName": "Silva",

"email": "marcos.henrique@toptal.com",

"password": "Y+XZEaR7J8xAQCc37nf1rw==$p8b5ykUx6xpC6k8MryDaRmXDxncLumU9mEVabyLdpotO66Qjh0igVOVerdqAh+CUQ4n/E0z48mp8SDTpX2ivuQ==",

"permissionLevel": 1,

"id": "5b02c5c84817bf28049e58a3"

}

exports.patchById = (req, res) => {

if (req.body.password){

let salt = crypto.randomBytes(16).toString('base64');

let hash = crypto.createHmac('sha512', salt).update(req.body.password).digest("base64");

req.body.password = salt + "$" + hash;

}

UserModel.patchUser(req.params.userId, req.body).then((result) => {

res.status(204).send({});

});

};