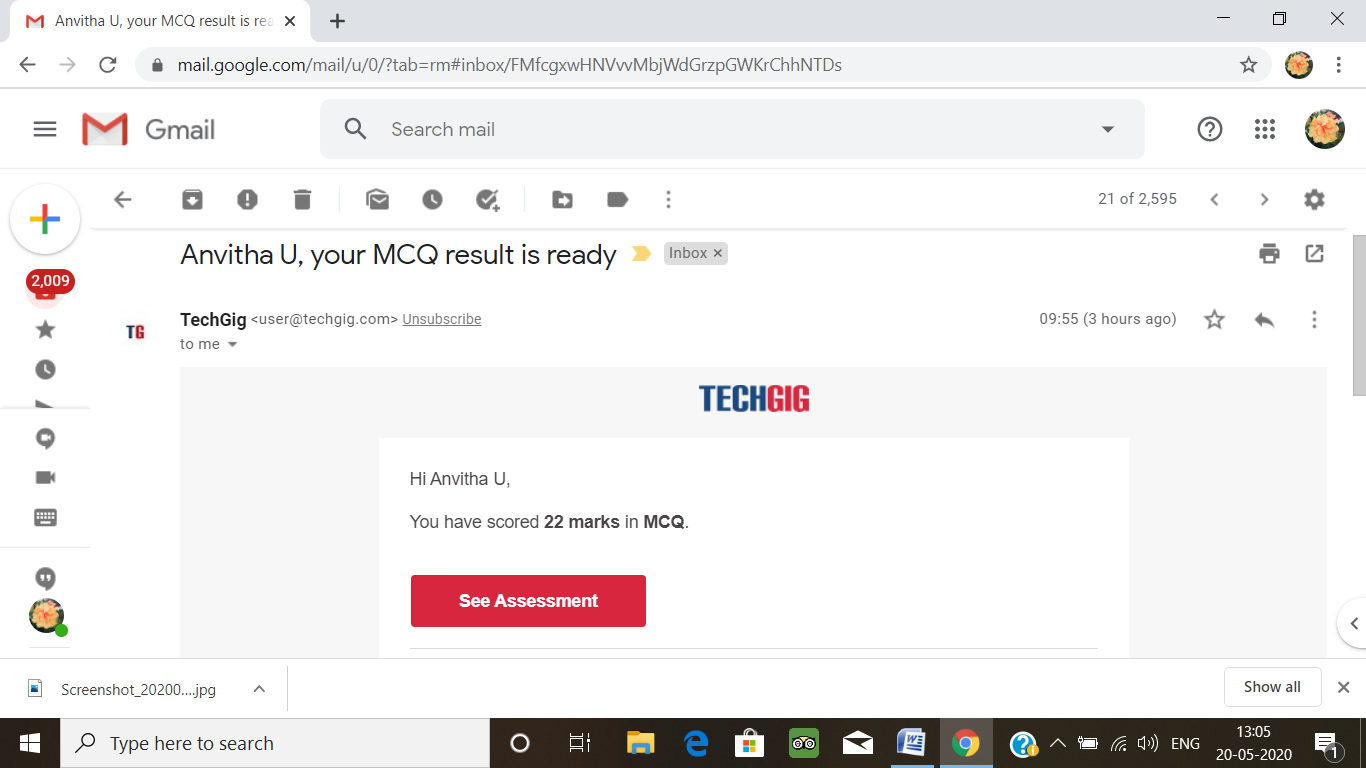
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **20-05-2020** | | | | | **Name:** | **Anvitha U** | |
| **Sem & Sec** | **A** | | | | | **USN:** | **4AL17CS009** | |
| **Online Test Summary** | | | | | | | | |
| **Subject** | | **SYSTEM SOFTWARE AND COMPILER DESIGN** | | | | | | |
| **Max. Marks** | | **30** | | **Score** | | | **22** | |
| **Certification Course Summary** | | | | | | | | |
| **Course** | **INTRODUCTION TO ETHICAL HACKING** | | | | | | | |
| **Certificate Provider** | | | Greatlearning  Academy | | **Duration** | | | 6hours |
| **Coding Challenges** | | | | | | | | |
| **Problem Statement:**1**.** Write a C Program to Reverse a Linked List in groups of given size.  2. Write a simple Python program to implement Diffie–Hellman Key Exchange Example  3. Write a Java Program which illustrates how to get column properties from ResultSet using ResultSetMetaData? | | | | | | | | |
| **Status: Done** | | | | | | | | |
| **Uploaded the report in Github** | | | | | **YES** | | | |
| **If yes Repository name** | | | | | <https://github.com/anvithauppoor/online_coding_activity> | | | |
| **Uploaded the report in slack** | | | | | **YES** | | | |

Online Test Details:

Subject:-System software and compiler design



Certification Course Details:

**Introduction to Ethical Hacking:**

Today I have studied **Ethical Hacking in Network Architecture Demonstration:**

Under this topic I have studied:-

\*OSI model.

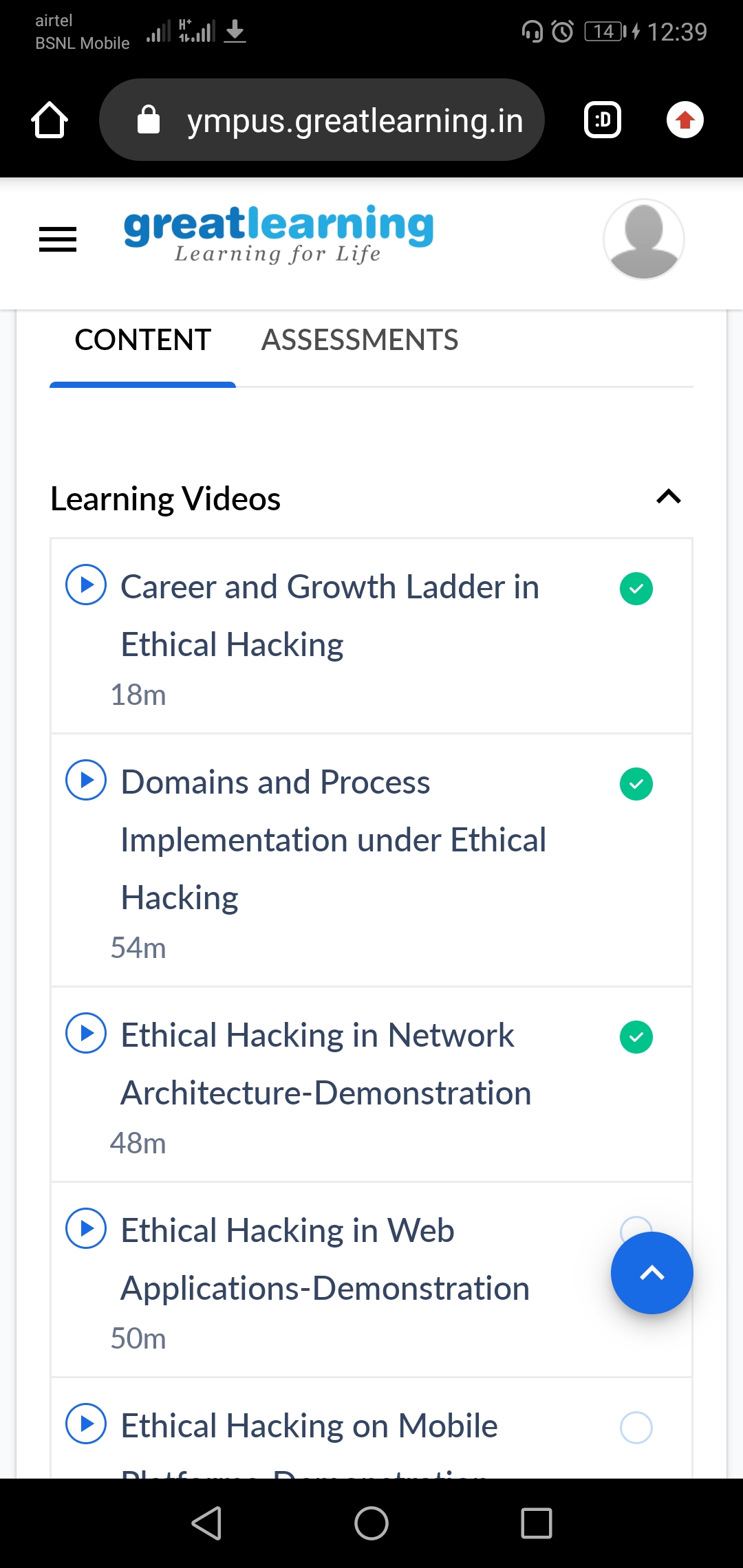
\*TCP/IP.

\*3 way handshaking.

\*Top tools for Networking Security.

\*Metasploit framework and architecture.

\*Demontration.



Coding Challenges Details:

1. 1. Write a C Program to Reverse a Linked List in groups of given size.

Test Case 1:  
If a linked listis: 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8  
The value of size k is 2  
Then the linked list looks like: 2 → 1 → 4 → 3 → 6 → 5 → 8 → 7

Test Case 2:  
If a linked listis: 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8  
The value of size k is 3  
Then the linked list looks like: 3 → 2 → 1 → 6 → 5 → 4 → 8 → 7

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

typedef struct node

{

int data;

struct node \*next;

}node;

void reverse(node \*head)

{

if(head == NULL)

return;

if(head -> next == NULL)

return;

reverse(head->next);

head->next->next = head;

head->next = NULL;

}

node \*swap\_in\_a\_group(node \*start , int k)

{

node \*p , \*q ,\*new\_start , \*temp;

int cnt;

p = start;

cnt = 0;

while(cnt != k-1)

{

if(p->next == NULL)

{

return start;

}

p = p->next;

cnt++;

}

new\_start = p;

q = new\_start;

while(1)

{

p = start;

temp = q->next;

if(temp == NULL)

{

reverse(p);

return new\_start;

}

q->next = NULL;

q = temp;

start = temp;

cnt = 0;

while(cnt != k-1)

{

if(temp->next == NULL)

{

reverse(p);

p->next = q;

return new\_start;

}

temp = temp->next;

cnt++;

}

reverse(p);

p->next = temp;

q = temp;

}

return new\_start;

}

int main()

{

int a , i , n , cnt , k=4 , flag = 1;

node \*p,\*q,\*start;

printf("Enter the number of nodes");

scanf("%d",&n);

printf("Enter all the nodes \n");

p = (node\*)malloc(sizeof(node));

scanf("%d",&a);

p->data = a;

p->next = NULL;

start = p;

for(i=1;i<n;i++)

{

q = (node\*)malloc(sizeof(node));

scanf("%d",&a);

q->data = a;

q->next = NULL;

p->next = q;

p = p->next;

}

printf("\n Enter K ");

scanf("%d",&k);

printf("\n swapped list==");

p = swap\_in\_a\_group(start , k);

while(p!=NULL)

{

printf("%d ",p->data);

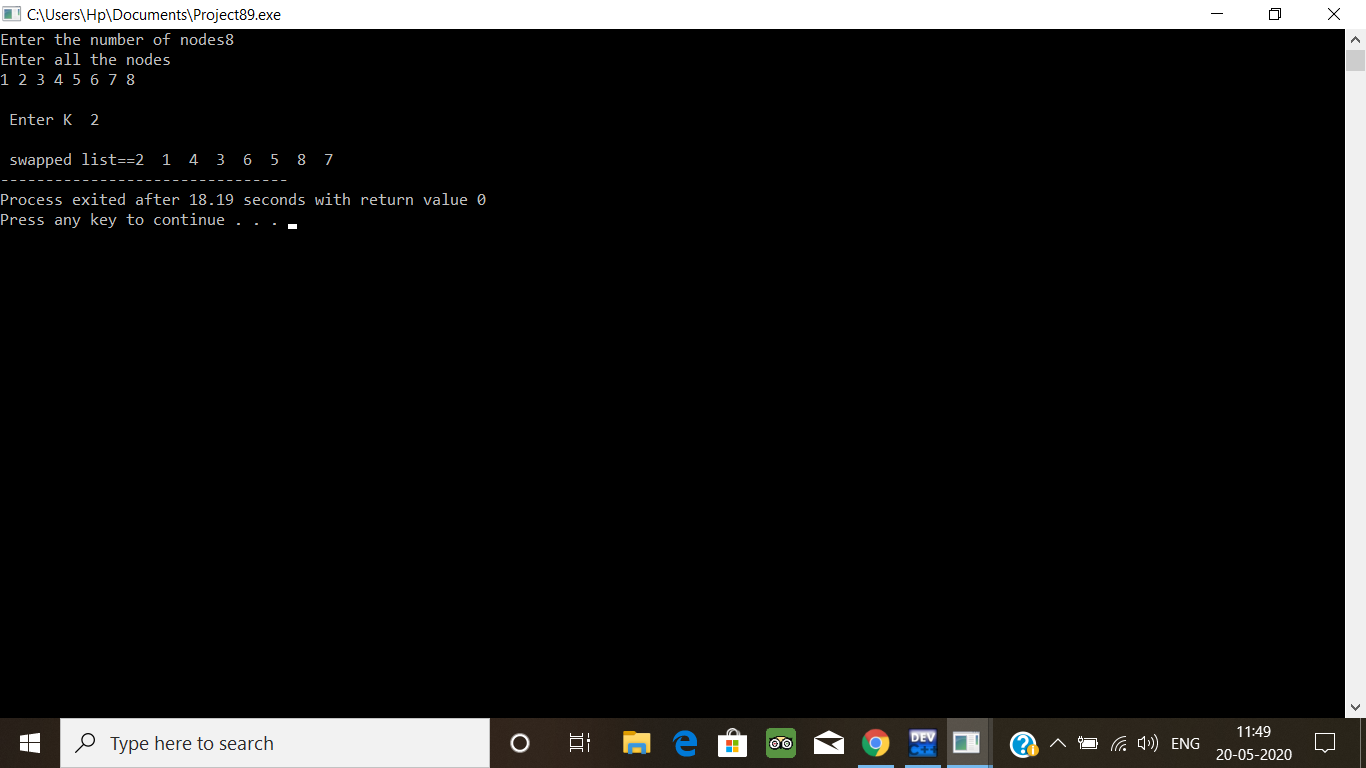
p = p->next;

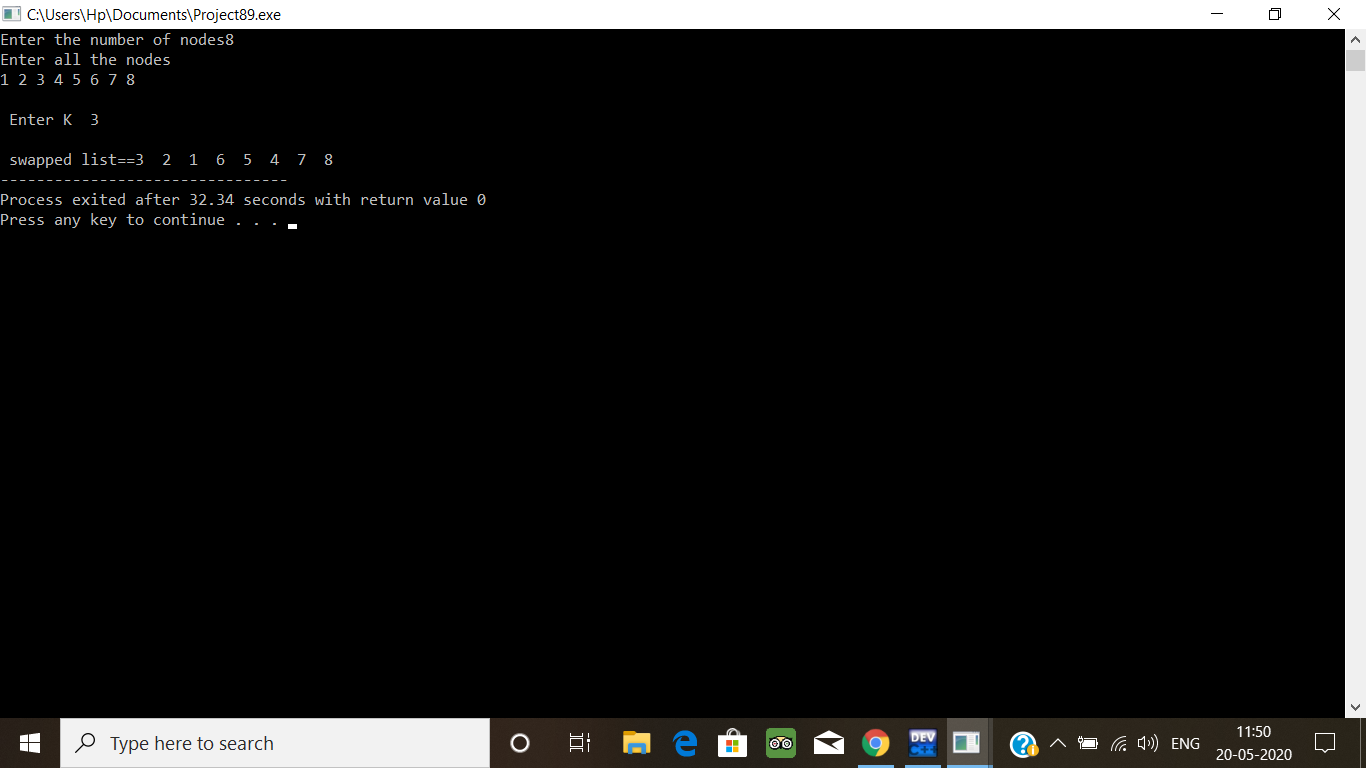
}

return 0;

}

**Output:**





2. .Write a simple Python program to implement Diffie–Hellman Key Exchange Example

sharedPrime = 23

sharedBase = 5

aliceSecret = 6

bobSecret = 15

print( "Publicly Shared Variables:")

print( " Publicly Shared Prime: " , sharedPrime )

print( " Publicly Shared Base: " , sharedBase )

A = (sharedBase\*\*aliceSecret) % sharedPrime

print( "\n Alice Sends Over Public Chanel: " , A )

B = (sharedBase \*\* bobSecret) % sharedPrime

print("\n Bob Sends Over Public Chanel: ", B )

print( "\n------------\n" )

print( "Privately Calculated Shared Secret:" )

# Alice Computes Shared Secret: s = B^a mod p

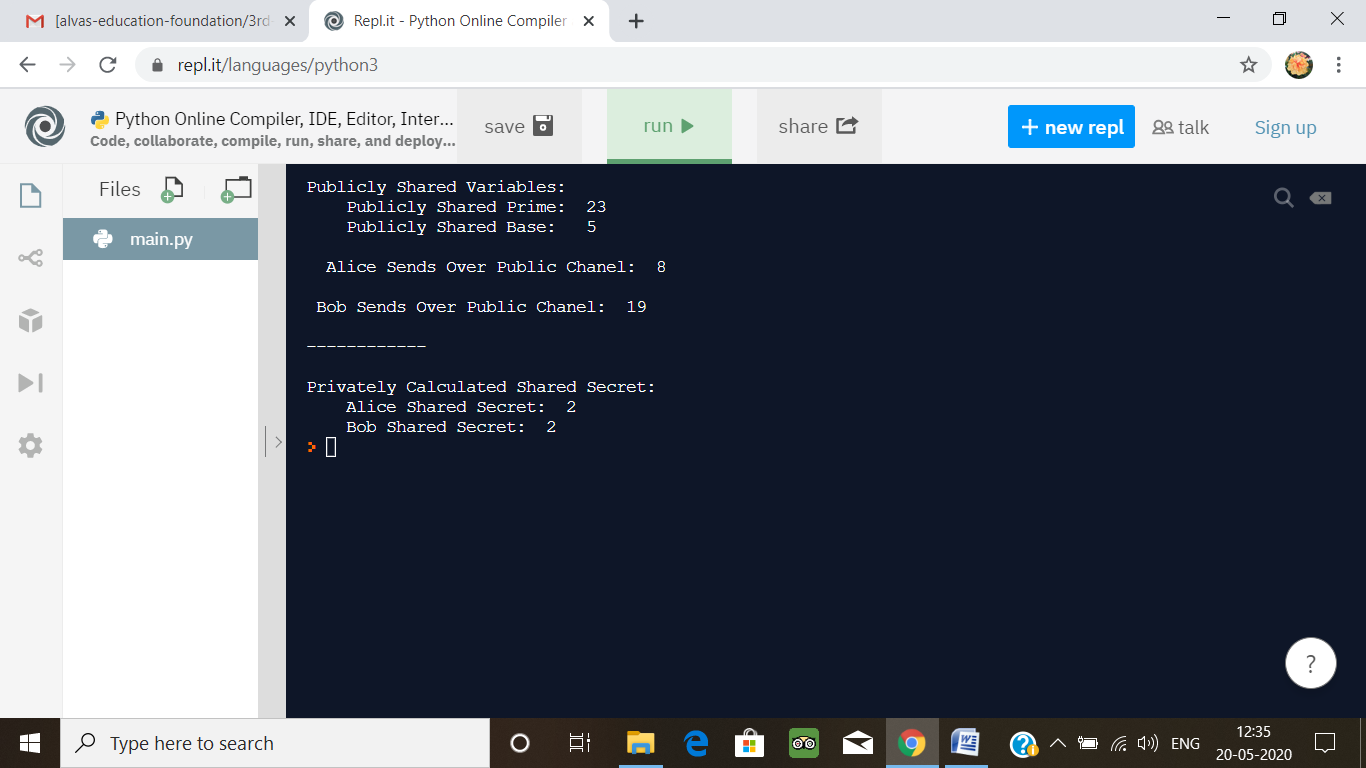
aliceSharedSecret = (B \*\* aliceSecret) % sharedPrime

print( " Alice Shared Secret: ", aliceSharedSecret )

bobSharedSecret = (A\*\*bobSecret) % sharedPrime

print( " Bob Shared Secret: ", bobSharedSecret )

**Output:**



3. Write a Java Program which illustrates how to get column properties from ResultSet using ResultSetMetaData?

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.ResultSetMetaData;

import java.sql.SQLException;

import java.sql.Statement;

public class Main {

public static void main(String a[]){

Connection con = null;

Statement st = null;

ResultSet rs = null;

try {

Class.forName("oracle.jdbc.driver.OracleDriver");

con = DriverManager.

getConnection("jdbc:oracle:thin:@<hostname>:<port num>:<DB name>"

,"user","password");

st = con.createStatement();

rs = st.executeQuery("select \* from emp");

ResultSetMetaData rsmd = rs.getMetaData();

int columnCount = rsmd.getColumnCount();

for(int i=0;i<=columnCount;i++){

System.out.println(rsmd.getColumnName(i));

System.out.println(rsmd.getColumnType(i));

}

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} finally{

try{

if(rs != null) rs.close();

if(st != null) st.close();

if(con != null) con.close();

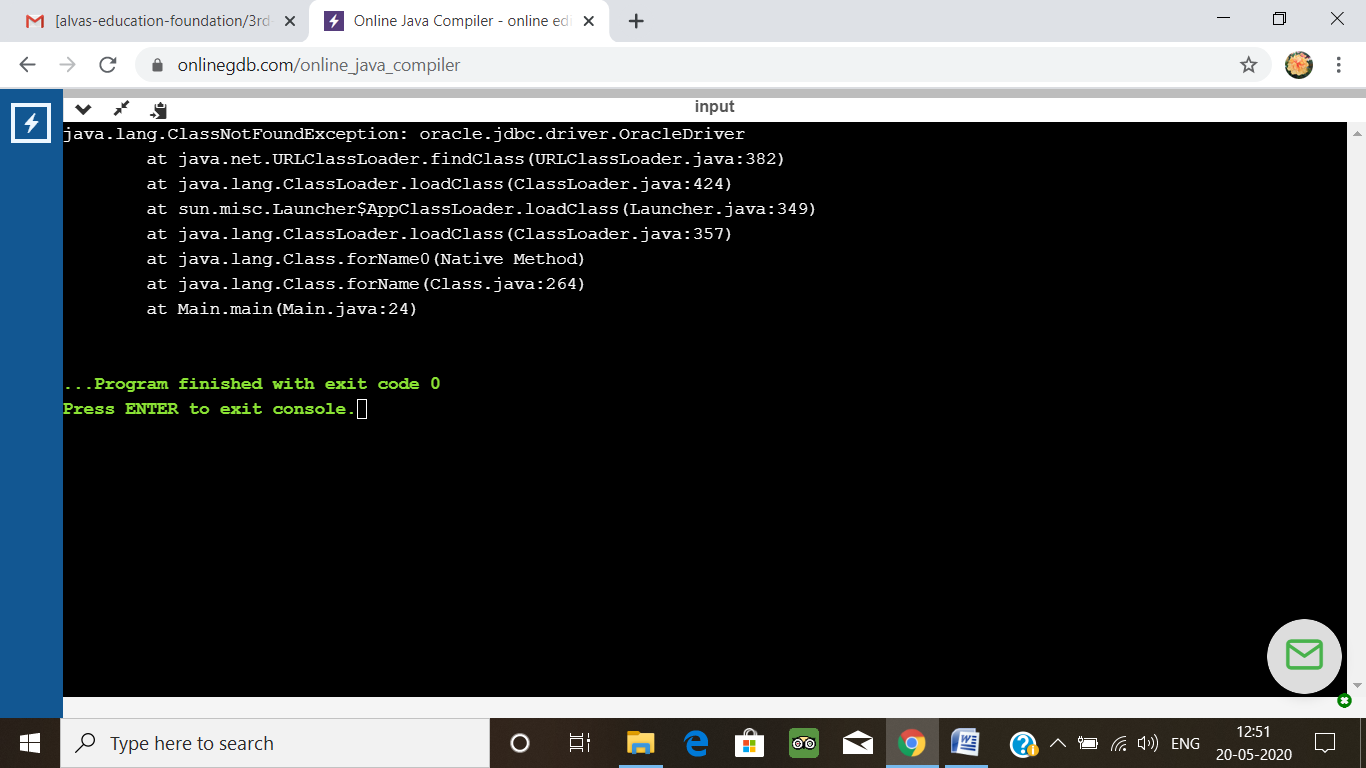
} catch(Exception ex){}

}

}

}

**Output:**

****