Bhumika Suhas Mane Roll No: 45

Index

Sr.No	Title of the Project	Date	Signature
1	Write programs to implement the following Substitution Cipher Techniques: a) Caesar Cipher b) Modified Caesar Cipher c) Monoalphabetic cipher		
2	Write programs to implement the following Substitution Cipher Techniques: a) Vigenere Cipher b) Playfair Cipher		
3	Write programs to implement the following Transposition Cipher Techniques: a) Rail Fence Cipher b) Vernam Cipher c) Simple Columnar Technique		
4	Write program to encrypt and decrypt strings using a) DES Algorithm b) AES Algorithm		
5	Write a program to implement RSA algorithm to perform encryption / decryption of a given string		
6	Write a program to implement the Diffie-Hellman Key Agreement algorithm to generate symmetric keys.		
7	Write a program to implement the MD5 algorithm compute the message digest.		
8	Write a program to calculate HMAC-SHA1 Signature		
9	Write a program to implement SSL.		
10	Configure Windows Firewall to block: a) A port b) A Program c) A website		

Bhumika Suhas Mane Roll No: 45

Date:21/06/2024

PRACTICAL 1

Write programs to implement the following Substitution Cipher Techniques:

a) Caesar Cipher
b) Modified Caesar Cipher
c) Monoalphabetic Cipher

a) caesercipher.java

```
package javaapplication1;
import java.util.Scanner;
public class CaesarCipher
{
public static final String
ALPHABET="abcdefghijklmnopqrstuvwxyz";
public static String encrypt(String plainText,int shiftKey)
{
plainText=plainText.toLowerCase();
String cipherText="";
for(int i=0;i<plainText.length();i++)
{
int charPosition =ALPHABET.indexOf(plainText.charAt(i));
int keyVal=(shiftKey+charPosition)%26;
char replaceVal=ALPHABET.charAt(keyVal);
cipherText+=replaceVal;
}
return cipherText;
}
public static String decrypt(String cipherText,int shiftKey)
{
cipherText=cipherText.toLowerCase();
String plainText="";
for(int i=0;i<cipherText.length();i++)
{
int charPosition=ALPHABET.indexOf(cipherText.charAt(i));
int keyVal=(charPosition-shiftKey)%26;
if(keyVal<0)
{
keyVal=ALPHABET.length()+keyVal;
```

```
}
char replaceVal=ALPHABET.charAt(keyVal);
plainText+=replaceVal;
}
return plainText;
}
public static void main(String[] args)
{
Scanner sc=new Scanner(System.in);
System.out.println("Enter the string for encryption: ");
String message=new String();
message=sc.next();
System.out.println(encrypt(message,3));
System.out.println(decrypt(encrypt(message,3),3));
sc.close();
} }
```

OUTPUT:

```
run:
Enter the string for encryption:
meetmeafterthetogaparty
phhwphdiwhuwkhwrjdsduwb
meetmeafterthetogaparty
BUILD SUCCESSFUL (total time: 21 seconds)
```

b) Modified Caesar Cipher

```
package javaapplication1;
import java.util.Scanner;
public class CaesarCipher
{
public static final String
ALPHABET="abcdefghijklmnopqrstuvwxyz";
public static String encrypt(String plainText,int shiftKey)
{
plainText=plainText.toLowerCase();
String cipherText="";
for(int i=0;i<plainText.length();i++)
{
int charPosition =ALPHABET.indexOf(plainText.charAt(i));
int keyVal=(shiftKey+charPosition)%26;
```

```
char replaceVal=ALPHABET.charAt(keyVal);
cipherText+=replaceVal;
}
return cipherText;
}
public static String decrypt(String cipherText,int shiftKey)
{
cipherText=cipherText.toLowerCase();
String plainText="";
for(int i=0;i<cipherText.length();i++)
{
int charPosition=ALPHABET.indexOf(cipherText.charAt(i));
int keyVal=(charPosition-shiftKey)%26;
if(keyVal<0)
{
keyVal=ALPHABET.length()+keyVal;
}
char replaceVal=ALPHABET.charAt(keyVal);
plainText+=replaceVal;
}
return plainText;
}
public static void main(String[] args)
{
Scanner sc=new Scanner(System.in);
System.out.println("Enter the key : ");
int shiftKey=sc.nextInt();
System.out.println("Enter the string for encryption: ");
String message=new String();
message=sc.next();
System.out.println(encrypt(message,shiftKey));
System.out.println(decrypt(encrypt(message,shiftKey),shiftKey));
sc.close();
} }
```

OUTPUT:

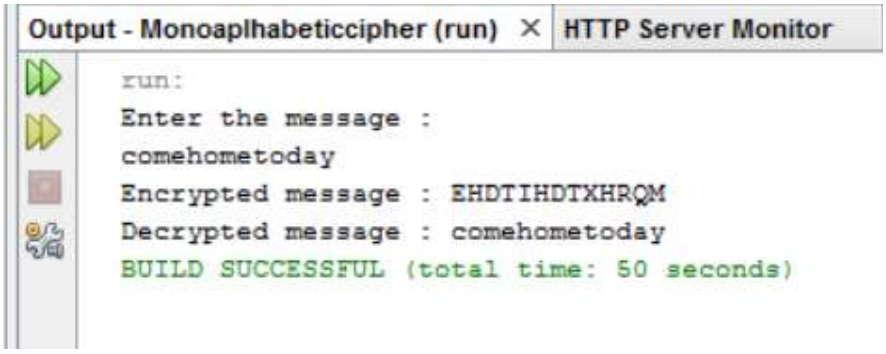
```
run:
Enter the key :
4
Enter the string for encryption:
come
gsqi
come
BUILD SUCCESSFUL (total time: 8 seconds)
```

c) monoaplhabeticcipher.java

```
package monoaplhabeticcipher;
import java.util.Scanner;
public class Monoaplhabeticcipher {
public static char
p[]={'a','b','c','d','e','f','g','h','i','j','k','l','m','n','n','o','p','q','r','s','t','u','v','w','x','y','z'};
public static char
ch[]={'Q','W','E','R','T','Y','U','I','O','P','A','S','D','F','G','H','J','K','L','Z','X','C','V','B','
N','M'};
public static String doEncryption(String s)
{
char c[]=new char[(s.length())];
for(int i=0;i<s.length();i++)
{
for(int j=0;j<26;j++)
{
if(p[j]==s.charAt(i))
{ c[i]=ch[j];
break; }
}
}
return(new String(c));
}
public static String doDecryption(String s)
{
char p1[]=new char[(s.length())];
for(int i=0;i<s.length();i++)
{
for(int j=0;j<26;j++)
{
if(ch[j]==s.charAt(i))
{ p1[i]=p[j];
break; }
}
}
return(new String(p1));
}
public static void main(String[] args) {
Scanner sc=new Scanner(System.in);
System.out.println("Enter the message : ");
String en=doEncryption(sc.next().toLowerCase());
System.out.println("Encrypted message : "+en);
System.out.println("Decrypted message : "+doDecryption(en));
```

```
sc.close();
}
}
```

OUTPUT:



Bhumika Suhas Mane Roll No: 45

Date: 05/07/2024

PRACTICAL 2

Write programs to implement the following Substitution Cipher Techniques:

a) Vigenere cipher
b) Playfair Cipher

a) vigenerecipher.java

```
package vigenerecipher;
public class VigenereCipher
{
static String generateKey(String str,String key)
{
int x=str.length();
for(int i=0;;i++)
{
if(x==i)
i=0;
if(key.length()==str.length())
break;
key+=(key.charAt(i));
}
return key;
}
static String cipherText(String str,String key)
{
String cipher_text="";
for(int i=0;i<str.length();i++)
{
int x=(str.charAt(i)+key.charAt(i))%26;
x+='A';
cipher_text+=(char)(x);
}
return cipher_text;
}
static String originalText(String cipher_text,String key)
{
String orig_text="";
for(int i=0;i<cipher_text.length()&&i<key.length();i++)
{
```

```
int x=(cipher_text.charAt(i)-key.charAt(i)+26)%26;
x+='A';
orig_text+=(char)(x);
}
return orig_text;
}
static String LowerToUpper(String s)
{
StringBuffer str=new StringBuffer(s);
for(int i=0;i<s.length();i++)
{
if(Character.isLowerCase(s.charAt(i)))
{
str.setCharAt(i, Character.toUpperCase(S.charAt(i)));
} }
s=str.toString();
return s;
}
public static void main(String[] args)
{
String Str="COME";
String Keyword="ABCA";
String str=LowerToUpper(Str);
String keyword=LowerToUpper(Keyword);
String key=generateKey(str,keyword);
String cipher_text=cipherText(str,key);
System.out.println("CipherText: "+cipher_text+"\n");
System.out.println("Original Decrypted Text:"+originalText(cipher_text,key));
 } }
```

OUTPUT:

```
put - vigenereCipher (run)

run:
CipherText: CPOE

Original Decrypted Text: COME
BUILD SUCCESSFUL (total time: 1 second)
```

b) playfaircipher.java

```
package playfaircipher;
import java.awt.Point;
import java.util.Scanner;
public class PlayfairCipher
{
private int length=0;
private String [][] table;
public static void main(String[] args)
{
PlayfairCipher pf=new PlayfairCipher();
}
private PlayfairCipher()
{
System.out.println("Enter the keys for playfair cipher:");
Scanner sc=new Scanner(System.in);
String key=parseString(sc);
while(key.equals(""))
key=parseString(sc);
table=this.cipherTable(key);
System.out.println("Enter the plaintext to be ciphertext: ");
String input=parseString(sc);
while(input.equals(""))
input=parseString(sc);
String output=cipher(input);
String decodedOutput=decode(output);
this.keyTable(table);
this.printResults(output,decodedOutput);
}
private String parseString(Scanner sc)
{
String parse= sc.nextLine();
parse=parse.toUpperCase();
parse=parse.replaceAll("[^A-Z]", "");
parse=parse.replace("J", "I");
return parse;
}
private String[][] cipherTable(String key)
{
String[][] playfairTable=new String[5][5];
String keyString=key+"ABCDEFGHKLMNOPQRSTUVWXYZ";
for(int i=0;i<5;i++)
for(int j=0;j<5;j++)
```

```
playfairTable[i][j]="";
for(int k=0; k <keyString.length();k++)
{
boolean repeat=false;
boolean used=false;
for(int i=0;i<5;i++)
{
for(int j=0;j<5;j++)
{
if(playfairTable[i][j].equals(""+keyString.charAt(k)))
{
repeat=true;
}
else if(playfairTable[i][j].equals("")&& !repeat && !used)
{
playfairTable[i][j]=""+keyString.charAt(k);
used=true;
} } } }
return playfairTable;
}
private String cipher(String in)
{
length=(int)in.length()/2+in.length()%2;
for(int i=0;i<(length-1);i++)
{
if(in.charAt(2*i)==in.charAt(2*i+1))
{
in=new StringBuffer(in).insert(2*i+1,'X').toString();
length=(int)in.length()/2+in.length()%2;
} }
String[] digraph=new String[length];
for(int j=0;j<length;j++)
{
if(j==(length-1)&&in.length()/2==(length-1))
in=in+"X";
digraph[j]=in.charAt(2*j)+""+in.charAt(2*j+1);
}
String out="";
String[] encDigraphs=new String[length];
encDigraphs=encodeDigraph(digraph);
for(int k=0;k<length;k++)
out=out+encDigraphs[k];
return out;
}
```

```
private String[] encodeDigraph(String di[])
{
String[] encipher=new String[length];
for(int i=0;i<length;i++)
{
char a=di[i].charAt(0);
char b=di[i].charAt(1);
int r1=(int) getPoint(a).getX();
int r2=(int) getPoint(b).getX();
int c1=(int) getPoint(a).getY();
int c2=(int) getPoint(b).getY();
if(r1==r2)
{
c1=(c1+1)%5;
c2=(c2+1)%5;
}
else if(c1==c2)
{
r1=(r1+1)%5;
r2=(r2+1)%5;
}
else
{
int temp=c1;
c1=c2;
c2=temp;
}
encipher[i]=table[r1][c1]+""+table[r2][c2];
}
return encipher;
}
private String decode(String out)
{
String decoded="";
for(int i=0;i<out.length()/2;i++)
{
char a=out.charAt(2*i);
char b=out.charAt(2*i+1);
int r1=(int) getPoint(a).getX();
int r2=(int) getPoint(b).getX();
int c1=(int) getPoint(a).getY();
int c2=(int) getPoint(b).getY();
if(r1==r2)
{
```

```
c1=(c1+4)%5;
c2=(c2+4)%5;
}
else if(c1==c2)
{
r1=(r1+4)%5;
r2=(r2+4)%5;
}
else
{
int temp=c1;
c1=c2;
c2=temp;
}
decoded=decoded+table[r1][c1]+table[r2][c2];
}
return decoded;
}
private Point getPoint(char c)
{
Point pt=new Point(0,0);
for(int i=0;i<5;i++)
for(int j=0;j<5;j++)
if(c==table[i][j].charAt(0))
pt=new Point(i,j);
return pt;
}
private void keyTable(String[][] printTable)
{
System.out.println("Playfair Cipher Key Matrix:");
System.out.println();
for(int i=0;i<5;i++)
{
for(int j=0;j<5;j++)
{
System.out.print(printTable[i][j]+"");
}
System.out.println();
}
System.out.println();
}
private void printResults(String encipher,String dec)
{
System.out.print("Encrypted Message: ");
```

```
System.out.println(encipher);
System.out.println();
System.out.print("Decrypted Message: ");
System.out.println(dec);
} }
```

OUTPUT:

```
Output - PlayfairCipher (run)
run:
Enter the keys for playfair cipher:
playfairexample
Enter the plaintext to be ciphertext:
mynameisatul
Playfair Cipher Key Matrix:

PLAYF
IREXM
BCDGH
KNOQS
TUVWZ

Encrypted Message: XFOLIXMKPVLR

Decrypted Message: MYNAMEISATUL
BUILD SUCCESSFUL (total time: 23 seconds)
```

Bhumika Suhas Mane Roll No: 45

Date: 12/07/2024

PRACTICAL 3

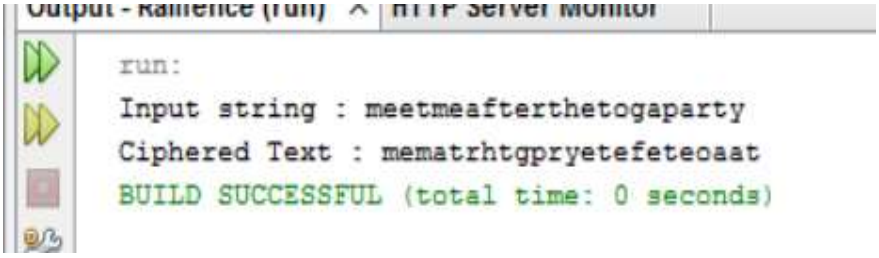
Write programs to implement the following Transposition Cipher Techniques:

a) Rail Fence Cipher
b) Vernam Cipher
c) Simple Columnar Technique

a) railfence.java

```
package railfence;
public class Railfence {
public static void main(String[] args) {
String input="meetmeafterthetogaparty";
String output="";
int len=input.length();
int flag=0;
System.out.println("Input string : "+input);
for(int i=0;i<len;i+=2)
{
output+=input.charAt(i);
}
for(int i=1;i<len;i+=2)
{
output+=input.charAt(i);
}
System.out.println("Ciphered Text : "+output);
} }
```

OUTPUT:



Bhumika Suhas Mane Roll No: 45

b) vernamecipher.java

```
package com.mycompany.vernamcipher;
import java.util.Scanner;
public class VernamCipher {
public static void main(String[] args) {
Scanner sc=new Scanner(System.in);
System.out.println("Enter String : ");
String txt=sc.nextLine();
System.out.println("Enter OTP(One-Time Pad): ");
String otp=sc.nextLine();
String st="";
char m,n;
int p1=0,p2=2;
char c[]=new
char[]{'a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z'};
int n1[]=new
int[]{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25};
if (txt.length()!=otp.length()){
System.out.println("Please enter OTP as the same length of string : ");
otp=sc.nextLine(); }
for(int i=0;i<txt.length();i++){
m=(char)(txt.charAt(i));
n=(char)(otp.charAt(i));
for(int j=0;j<c.length;j++){
if(m==c[j]){p1=n1[j];}
if(n==c[j]){p2=n1[j];}
}
int p=p1+p2;
System.out.println(p1+"+"+p2+"=");
System.out.println(p);
if(p>=26){p=p-26;}
char c1=c[p];
System.out.println("\n\tCHARACTER at "+p+" is "+c1);
st=st+c1;
}
System.out.println("_________________");
System.out.println("Cipher text is : "+st);
}
}
```

Bhumika Suhas Mane Roll No: 45

OUTPUT:

```
run:
Enter String :
howareyou
Enter OTP(One-Time Pad):
ncbtzqarx
7+13=
20

        CHARACTER at 20 is u
14+2=
16

        CHARACTER at 16 is q
22+1=
23

        CHARACTER at 23 is x
0+19=
19

        CHARACTER at 19 is t
17+25=
42

        CHARACTER at 16 is q
4+16=
20

        CHARACTER at 20 is u
24+0=
24

        CHARACTER at 24 is y
14+17=
31

        CHARACTER at 5 is f
20+23=
43

        CHARACTER at 17 is r
Cipher text is : uqxtquyfr
BUILD SUCCESSFUL (total time: 55 seconds)
```

c) simplecolumnarcipher.java

```
import java.util.*;
public class Simplecolumnarcipher {
public static void main(String[] args) {
Scanner sc=new Scanner(System.in);
System.out.print("Enter plaintext : ");
String message=sc.nextLine();
System.out.println("Enter key in number : ");
String key=sc.nextLine();
int columnCount=key.length();
int rowCount=(message.length()+columnCount-1)/columnCount;
int plainText[][]=new int[rowCount][columnCount];
int cipherText[][]=new int[rowCount][columnCount];
System.out.print("\n Encryption");
```

```
cipherText=encrypt(plainText,cipherText,message,rowCount,columnCount,key
);
String ct="";
for(int i=0;i<columnCount;i++)
{
for(int j=0;j<rowCount;j++)
{
if(cipherText[j][i]==0) ct=ct+'x';
else{ ct=ct+(char)cipherText[j][i]; }
}
}
System.out.print("\n Cipher Text : "+ct.toString());
System.out.print("\n Decryption");
plainText=decrypt(plainText,cipherText,ct,rowCount,columnCount,key);
String pt="";
for(int i=0;i<rowCount;i++)
{
for(int j=0;j<columnCount;j++)
{ if(plainText[i][j]==0) pt=pt+"";
else{ pt=pt+(char)plainText[i][j]; }
}
}
System.out.print("Plain text : "+pt);
System.out.println();
}
static int[][]encrypt(int plainText[][],int cipherText[][],String message,int
rowCount,int columnCount,String key)
{
int i,j;
int k=0;
for(i=0;i<rowCount;i++)
{
for(j=0;j<columnCount;j++)
{
if(k<message.length())
{
plainText[i][j]=(int)message.charAt(k);
k++;
}
else
{ plainText[i][j]='x'; }
}
}
for(i=0;i<columnCount;i++)
```

```
{
int currentCol=((int)key.charAt(i)-48)-1;
for(j=0;j<rowCount;j++)
{ cipherText[j][i]=plainText[j][currentCol]; }
}
System.out.print("Cipher array \n");
for(i=0;i<rowCount;i++)
{
for(j=0;j<columnCount;j++)
{
System.out.print((char)cipherText[i][j]+"");
}
System.out.println();
}
return cipherText;
}
static int[][]decrypt(int plainText[][],int cipherText[][],String message,int
rowCount,int columnCount,String key)
{ int i,j; for(i=0;i<columnCount;i++)
{
int currentCol=((int)key.charAt(i)-48)-1;
for(j=0;j<rowCount;j++)
{ plainText[j][currentCol]=cipherText[j][i]; }
}
System.out.print("Plain array \n");
for(i=0;i<rowCount;i++)
{
for(j=0;j<columnCount;j++)
{
System.out.print((char)plainText[i][j]+"\t");
}
System.out.println();
}
return plainText;
} }
```

Bhumika Suhas Mane Roll No: 45

OUTPUT:

```
Enter plaintext : attackpostponeduntiltwoam
Enter key in number :
4312567

 EncryptionCipher array
atatckp
ptosone
tnduilt
mawoxxx

 Cipher Text : aptmttnaaodwtsuocoixknlxpetx
 DecryptionPlain array
a       t       t       a       c       k       p
o       s       t       p       o       n       e
d       u       n       t       i       l       t
w       o       a       m       x       x       x
Plain text : attackpostponeduntiltwoamxxx
```

```
Enter plaintext : comehometomorrow
Enter key in number :
461253

 EncryptionCipher array
eocohm
oomemt
wxrrxo

 Cipher Text : eowooxcmroerhmxmto
 DecryptionPlain array
c       o       m       e       h       o
m       e       t       o       m       o
r       r       o       w       x       x
Plain text : comehometomorrowxx
-------------------------------------------------
BUILD SUCCESS
```

Bhumika Suhas Mane Roll No: 45

Date: 03/08/2024

PRACTICAL 4

Write program to encrypt and decrypt strings using

a) DES Algorithm

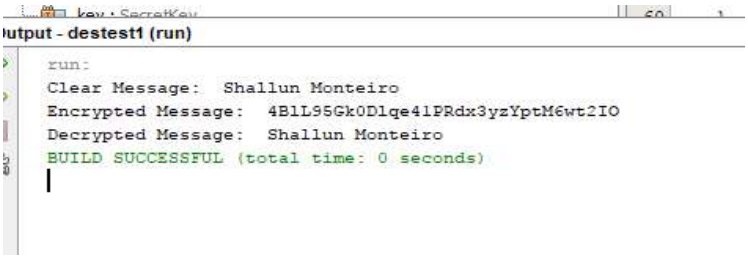
b) AES Algorithm

a) destest1

```
package destest1;
import com.sun.crypto.provider.DESKeyFactory;
import javax.crypto.Cipher;
import javax.crypto.SecretKeyFactory;
import javax.crypto.SecretKey;
import javax.crypto.spec.DESKeySpec;
import sun.misc.BASE64Decoder;
import sun.misc.BASE64Encoder;
public class Destest1
{
private SecretKey key;
public String theKey;
public void generateKey() throws Exception{
DESKeySpec deskeySpec = new DESKeySpec(theKey.getBytes());
SecretKeyFactory keyFactory =SecretKeyFactory.getInstance("DES");
key = keyFactory.generateSecret(deskeySpec);
}
public String encrypt(String messg) throws Exception{
Cipher cipher = Cipher.getInstance("DES");
cipher.init(cipher.ENCRYPT_MODE,key);
byte[]stringBytes=messg.getBytes("UTF-8");
byte[]raw = cipher.doFinal(stringBytes);
BASE64Encoder encode = new BASE64Encoder();
String base64 = encode.encode(raw);
return base64;
}
public String decrypt(String encrypted) throws Exception{
Cipher cipher = Cipher.getInstance("DES");
cipher.init(cipher.DECRYPT_MODE,key);
BASE64Decoder decode = new BASE64Decoder();
byte[]raw = decode.decodeBuffer(encrypted);
byte[]stringBytes = cipher.doFinal(raw);
String clear = new String(stringBytes,"UTF-8");
return clear;
```

```
}
public static void main(String[] args)
{
String messg = "Shallun Monteiro";
String decrypted;
String encrypted;
Destest1 des = new Destest1();
des.theKey = "1,2,3,4,5,6";
try{
des.generateKey();
System.out.println("Clear Message:  "+messg);
encrypted = des.encrypt(messg);
decrypted = des.decrypt(encrypted);
System.out.println("Encrypted Message:  "+encrypted);
System.out.println("Decrypted Message:  "+decrypted);
}
catch(Exception e){
}
} }
```

OUTPUT:



b) aestest

```
package aestest;
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;
import java.security.Key;
import sun.misc.BASE64Decoder;
import sun.misc.BASE64Encoder;
public class Aestest
{
private byte[] keyValue;
public Aestest(String key) {
keyValue = key.getBytes();
}
private Key generateKey() throws Exception {
```

```
Key key = new SecretKeySpec(keyValue, "AES");
return key; }
public String encrypt(String messg) throws Exception {
Key key = generateKey();
Cipher cipher = Cipher.getInstance("AES");
cipher.init(Cipher.ENCRYPT_MODE, key);
byte[] raw = cipher.doFinal(messg.getBytes());
BASE64Encoder encoder = new BASE64Encoder();
String base64 = encoder.encode(raw);
return base64;
}
public String decrypt(String encrypted) throws Exception {
Key key = generateKey();
Cipher cipher = Cipher.getInstance("AES");
cipher.init(Cipher.DECRYPT_MODE, key);
BASE64Decoder decoder = new BASE64Decoder();
byte[] raw = decoder.decodeBuffer(encrypted);
byte[] stringBytes = cipher.doFinal(raw);
String clear = new String(stringBytes, "UTF8");
return clear;
}
public static void main(String[] args) {
String messg = "MITTU DON";
String decrypted;
String encrypted;
Aestest aest = new Aestest("1v39eptlvuhaqqsr");
try {
System.out.println("AES:");
System.out.println("Clear Message: " + messg);
encrypted = aest.encrypt(messg);
System.out.println("Encrypted Message: " + encrypted);
decrypted = aest.decrypt(encrypted);
System.out.println("Decrypted Message: " + decrypted);
} catch (Exception e)
{ e.printStackTrace(); } } }
```

OUTPUT:

```
run:
AES:
Clear Message: MITTU DON
Encrypted Message: NybMTwPDDa73tlze7nfaKg==
Decrypted Message: MITTU DON
BUILD SUCCESSFUL (total time: 0 seconds)
```

Bhumika Suhas Mane Roll No: 45

Date: 27/07/2024

PRACTICAL 5

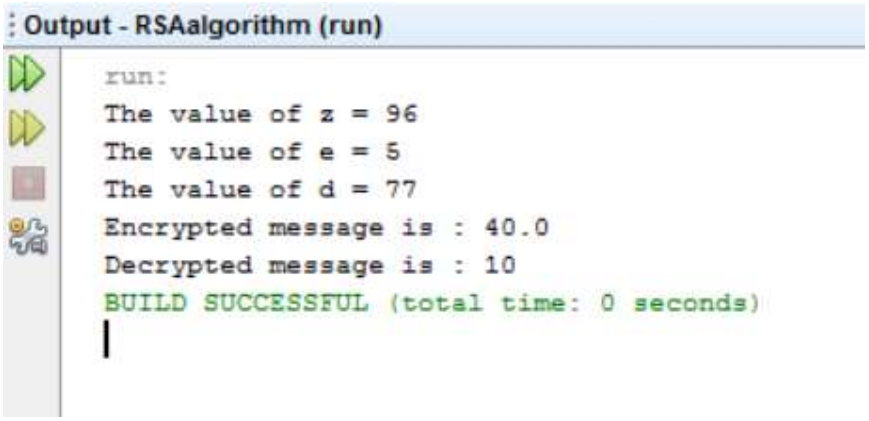
Write a program to implement RSA algorithm to perform encryption / decryption of a given string.

rsaalgorithm.java

```
package rsaalgorithm;
import java.math.*;
import java.util.*;
public class RSAalgorithm {
public static void main(String[] args) {
int p,q,n,phi,d=0,e,i;
int msg=10;
double c;
BigInteger msgback;
p=7;
q=17;
n=p*q;
phi=(p-1)*(q-1);
System.out.println("The value of z = "+phi);
for(e=2;e<phi;e++)
{ if(gcd(e,phi)==1)
{ break; }
}
System.out.println("The value of e = "+e);
for(i=0;i<=9;i++)
{
int x=1+(i*phi);
if(x%e==0)
{ d=x/e;
break; }
}
System.out.println("The value of d = "+d);
c=(Math.pow(msg,e))%n;
System.out.println("Encrypted message is : "+c);
BigInteger N=BigInteger.valueOf(n);
BigInteger C=BigDecimal.valueOf(c).toBigInteger();
msgback=(C.pow(d)).mod(N);
System.out.println("Decrypted message is : "+msgback);
}
```

```
static int gcd(int e,int z) {
if(e==0)
return z;
else
return gcd(z%e,e);
}
}
```

OUTPUT:



Bhumika Suhas Mane Roll No: 45

Date:27/07/2024

PRACTICAL 6

Write a program to implement the Diffie-Hellman Key Agreement algorithm to generate symmetric keys.

diffiehelmen.java

```
package diffiehellman;
public class DiffieHellman
{
private static long power(long a,long b,long p)
{
if(b==1)
return a;
else
return (((long)Math.pow(a,b))%p);
}
public static void main(String[] args)
{
long n,g,x,A,y,B,Ka,Kb;
n=11;
System.out.println("The value of N : "+n);
g=7;
System.out.println("The value of g : "+g);
x=3;
System.out.println("The private key for Alice : "+x);
A=power(g,x,n);
System.out.println("Value of A --> "+A);
y=6;
System.out.println("The private key for Bob : "+y);
B=power(g,y,n);
System.out.println("Value of B --> "+B);
Ka=power(B,x,n);
Kb=power(A,y,n);
System.out.println("Secret key for Alice is : "+Ka);
System.out.println("Secret key for Bob is : "+Kb);
}
}
```

Bhumika Suhas Mane Roll No: 45

OUTPUT:

```
Output - DiffieHellman (run)
run:
The value of N : 11
The value of g : 7
The private key for Alice : 3
Value of A --> 2
The private key for Bob : 6
Value of B --> 4
Secret key for Alice is : 9
Secret key for Bob is : 9
BUILD SUCCESSFUL (total time: 0 seconds)
```

Bhumika Suhas Mane Roll No: 45

Date:17/08/2024

PRACTICAL 7

Write a program to implement the MD5 algorithm compute the message digest.

mdhash.java

```
package mdhash;
import java.math.BigInteger;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
public class MDHash
{
   public static void main(String[] args)
   {
      System.out.println("MD Algorithm");
      System.out.println("For null "+md5(""));
      System.out.println("For Simple text"+md5("This is my text"));
      System.out.println("For Simple numbers"+md5("12345"));
   }
   public static String md5(String input)
   {
      String md5=null;
      if(null==input) return null;
      try
      {
         MessageDigest digest=MessageDigest.getInstance("MD5");
         digest.update(input.getBytes(),0,input.length());
         md5=new BigInteger(1,digest.digest()).toString(16);
      }
      catch(NoSuchAlgorithmException e)
      {  e.printStackTrace();  }
      return md5;
   } }
```

OUTPUT:

```
run:
MD Algorithm
For null d41d8cd98f00b204e9800998ecf8427e
For Simple text88b19be96ab393523e1553cf8e871e4
For Simple numbers827ccb0eea8a706c4c34a16891f84e7b
BUILD SUCCESSFUL (total time: 0 seconds)
```

Bhumika Suhas Mane Roll No: 45

Date:17/08/2024

PRACTICAL 8

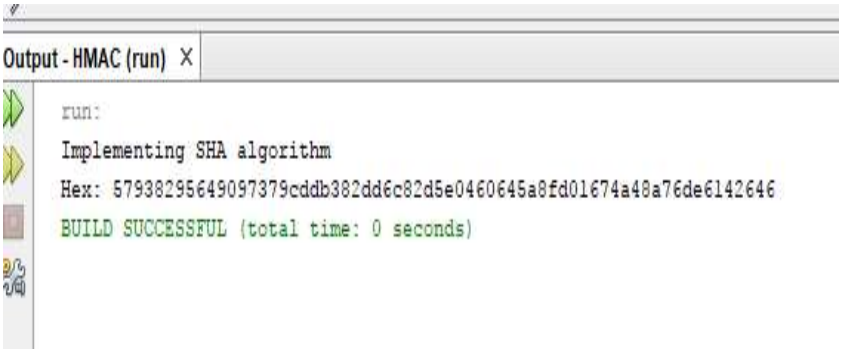
Write a program to calculate HMAC-SHA1 Signature.

hmac.java

```
package hmac;
import java.io.UnsupportedEncodingException;
import java.math.BigInteger;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
public class HMAC
{
static public byte[] calcHmacSha256(byte[] secretKey,byte[] message)
{
byte[] hmacSha256=null;
try
{
Mac mac=Mac.getInstance("HmacSHA256");
SecretKeySpec secretKeySpec=new
SecretKeySpec(secretKey,"HmacSHA256");
mac.init(secretKeySpec);
hmacSha256=mac.doFinal(message);
}
catch(Exception e)
{
throw new RuntimeException("Failed to calculate hmac-sha256",e);
}
return hmacSha256;
}
public static void main(String[] args)
{
try
{
byte[] hmacSha256;
hmacSha256=HMAC.calcHmacSha256("secret123".getBytes("UTF-8"),"hello
world".getBytes("UTF-8"));
System.out.println("Implementing SHA algorithm");
System.out.println(String.format("Hex: %032x",new
BigInteger(1,hmacSha256)));
}
catch(UnsupportedEncodingException e)
{
```

```
e.printStackTrace();
}
}
}
```

OUTPUT:



Bhumika Suhas Mane Roll No: 45

Date:14/09/2024

PRACTICAL 9

Write a program to implement SSL.

SSLServer.java

```
package sslserver;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.net.ssl.SSLServerSocketFactory;
public class SSLServer
{
static final int port=8000;
public static void main(String[] args)
{
SSLServerSocketFactory
sslServerSocketFactory=(SSLServerSocketFactory)SSLServerSocketFactory.
getDefault();
try
{
ServerSocket
sslServerSocket=sslServerSocketFactory.createServerSocket(port);
System.out.println("SSL ServerSocket Started");
System.out.println(sslServerSocket.toString());
Socket socket=sslServerSocket.accept();
System.out.println("ServerSocket Accepted");
PrintWriter out=new PrintWriter(socket.getOutputStream(),true);
try(BufferedReader bufferedReader=new BufferedReader(new
InputStreamReader(socket.getInputStream())))
{
String line;
while((line=bufferedReader.readLine())!=null)
{
System.out.println(line);
```

```
out.println(line);
} }
System.out.println("Closed");
}
catch(IOException ex)
{
Logger.getLogger(SSLServer.class.getName()).log(Level.SEVERE, null, ex);
} }  }
```

SSLClient.java

```
package sslclient;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.Socket;
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.net.ssl.SSLServerSocketFactory;
import javax.net.ssl.SSLSocketFactory;
public class SSLClient
{
static final int port=8000;
public static void main(String[] args)
{
SSLSocketFactory
sslSocketFactory=(SSLSocketFactory)SSLSocketFactory.getDefault();
try
{
Socket socket=sslSocketFactory.createSocket("localhost",port);
PrintWriter out=new PrintWriter(socket.getOutputStream(),true);
try(BufferedReader bufferedReader=new BufferedReader(new
InputStreamReader(socket.getInputStream())))
{
Scanner scanner=new Scanner(System.in);
while(true)
{
System.out.println("Enter something ");
String inputLine=scanner.nextLine();
if(inputLine.equals("q"))
```

Bhumika Suhas Mane Roll No: 45

```
{ break; }
out.println(inputLine);
System.out.println(bufferedReader.readLine());
} } }
catch(IOException ex)
{
Logger.getLogger(SSLClient.class.getName()).log(Level.SEVERE, null, ex);
} }  }
```

OUTPUT:

```
Microsoft Windows [Version 10.0.16299.125]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Student>keytool -genkey -alias signFiles -keystore testStr
Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]:  bhu
What is the name of your organizational unit?
  [Unknown]:  model
What is the name of your organization?
  [Unknown]:  model
What is the name of your City or Locality?
  [Unknown]:  dom
What is the name of your State or Province?
  [Unknown]:  maha
What is the two-letter country code for this unit?
  [Unknown]:  in
Is CN=bhu, OU=model, O=model, L=dom, ST=maha, C=in correct?
  [no]:  yes

Enter key password for <signFiles>
        (RETURN if same as keystore password):
Re-enter new password:

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard format using "keytool -importkeystore -srckeystore test
Str -destkeystore testStr -deststoretype pkcs12".
```

```
C:\Users\Student>java -jar -Djavax.net.ssl.keyStore=testStr -Djavax.net.ssl.keyStorePassword=bhumika "C:\Users\Student\Documents\NetBeansProjects\SSLServer\dist\SSLServ
er.jar"
SSL ServerSocket Started
[SSL: ServerSocket[addr=0.0.0.0/0.0.0.0,localport=8000]]
ServerSocket Accepted
hi
```

Command Prompt - java -jar -Djavax.net.ssl.trustStore=testStr -Djavax.net.ssl.trustStorePassword=bhumika "C:\Users\Student\Documents\NetBeansPr...

```
Microsoft Windows [Version 10.0.16299.125]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Student>java -jar -Djavax.net.ssl.trustStore=testStr -Djavax.net.ssl.trustStorePassword=bhumika "C:\Users\Stude
nt\Documents\NetBeansProjects\SSLClient\dist\SSLClient.jar"
Enter something
hi
hi
Enter something
```

Bhumika Suhas Mane Roll No: 45

Date:31/08/2024

PRACTICAL 10

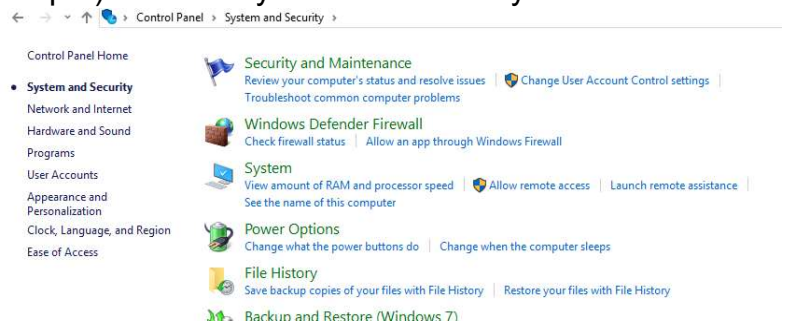
Configure Windows Firewall to block:

a) A port
b) A Program
c) A website

a) A Port

Step 1) Open control panel

Step 2) Click on system and security

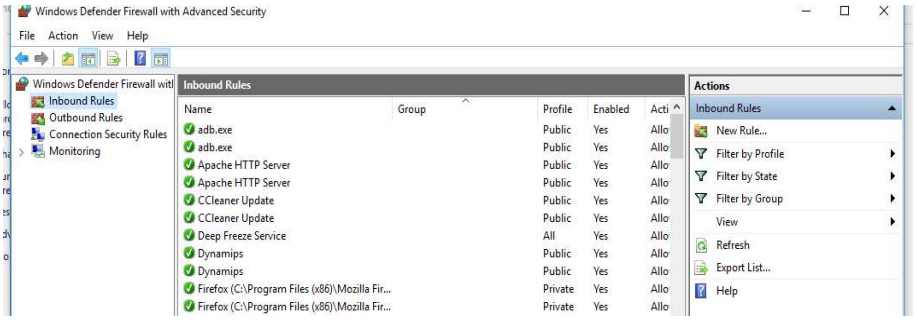


Step 3) Open Windows Defender Firewall

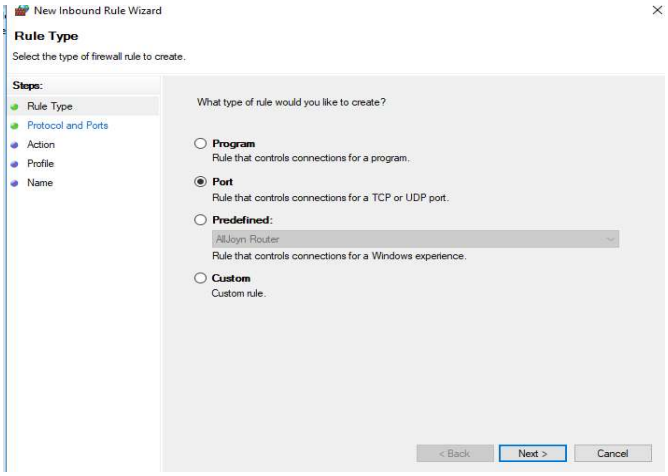
Step 4) Click on advanced settings

Bhumika Suhas Mane Roll No: 45

Step 5) Click on inbound rules and then on right side ->click on new rule

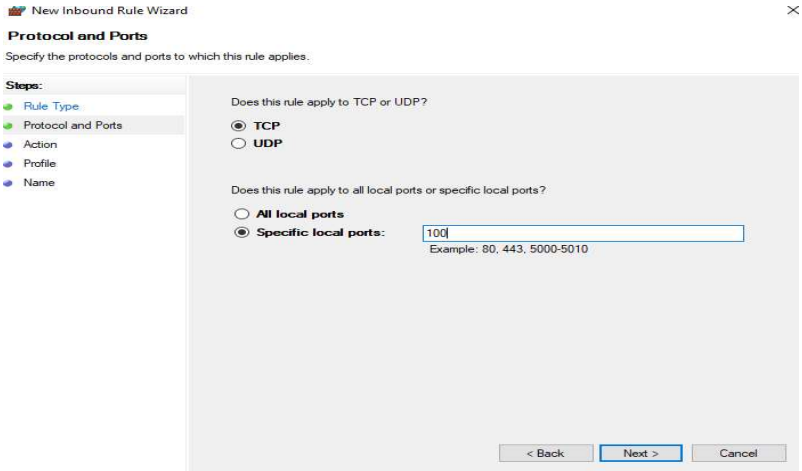


Step 6) Now Select port option and then click on next

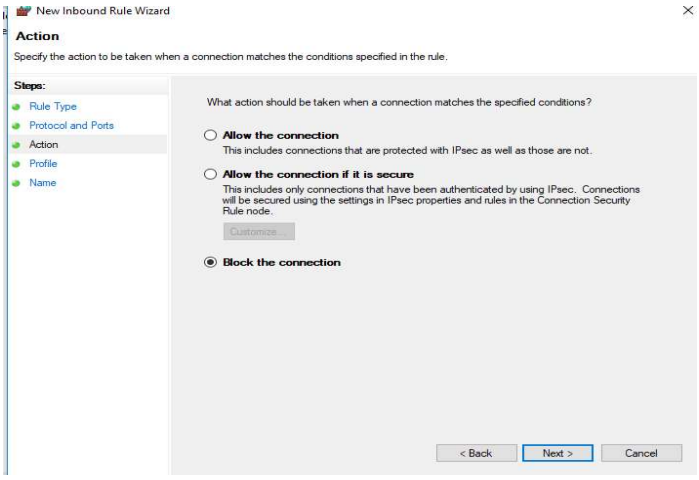


Step 7) Type 100 as specific local port, then click next

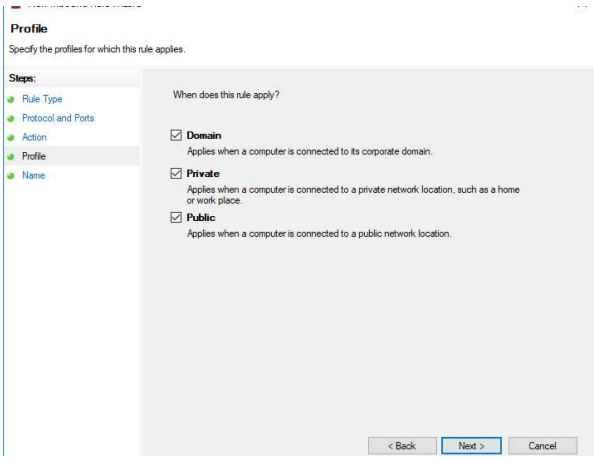
Bhumika Suhas Mane Roll No: 45



Step 8) Now select block the connection and click on next

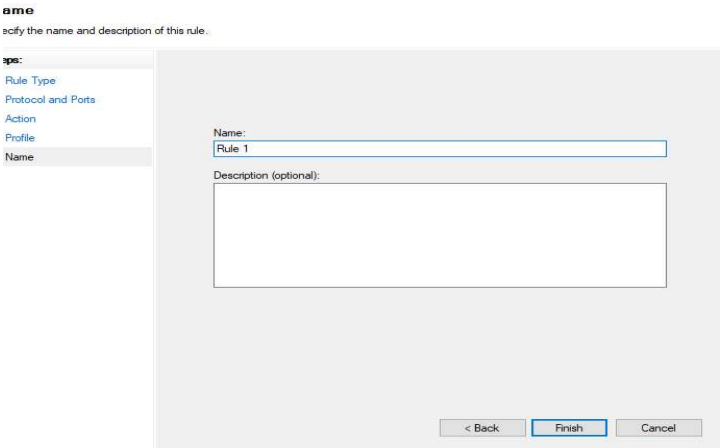


Step 9) Here, let the default changes be as it is and click on next

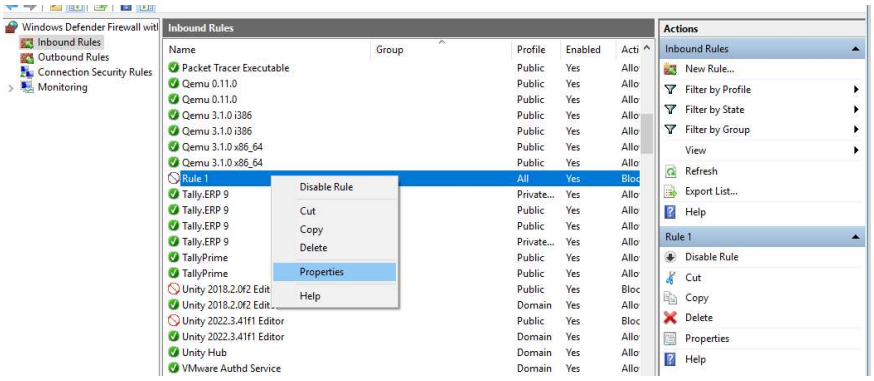


Bhumika Suhas Mane Roll No: 45

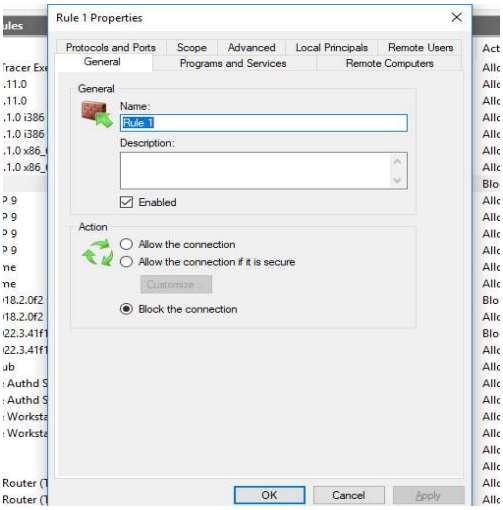
Step 10) Now type name as Rule 1 (the name by which we can identify our protocol) and then click on finish.



Step 11) Now right click on the Rule 1 and go to properties option.

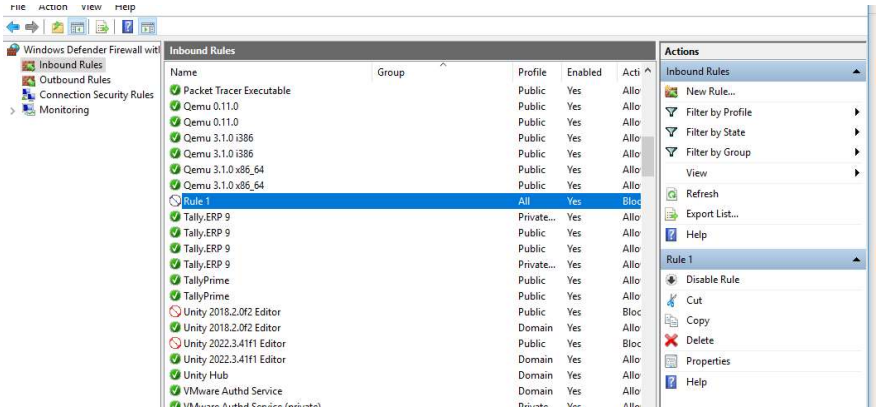


Step 12) Don't change anything just click on ok



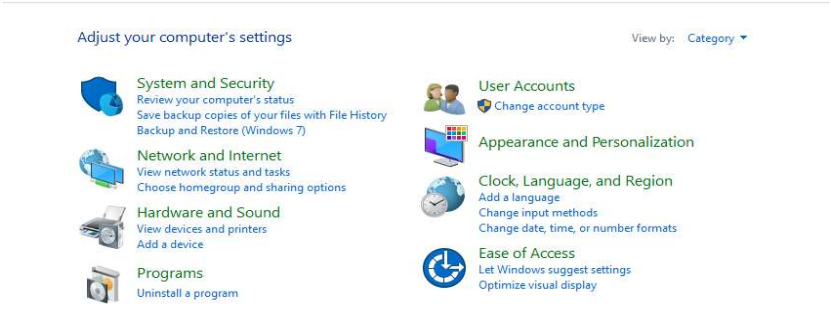
Bhumika Suhas Mane Roll No: 45

Step 13) We can also change the properties or disable the protocol using properties OR through the right side panel.

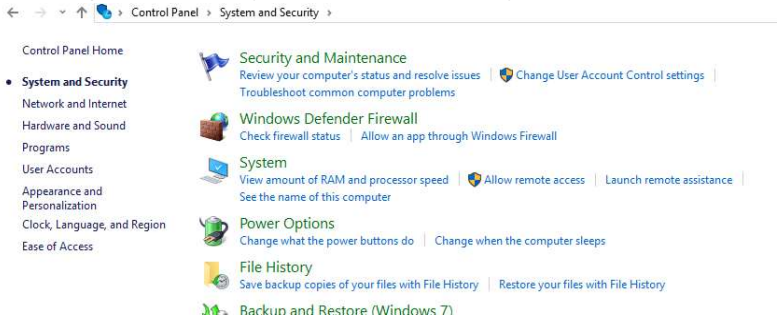


b) A Program

Step 1) Open control panel



Step 2) Click on system and security

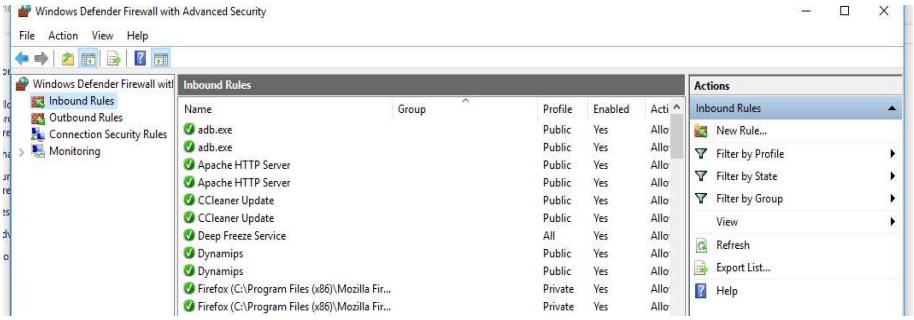


Step 3) Open Windows Defender Firewall

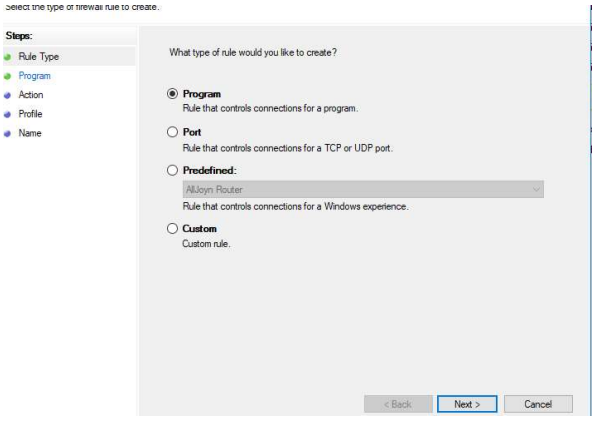
Bhumika Suhas Mane Roll No: 45

Step 4) Click on advanced settings

Step 5) Click on inbound rules and then on right side ->click on new rule

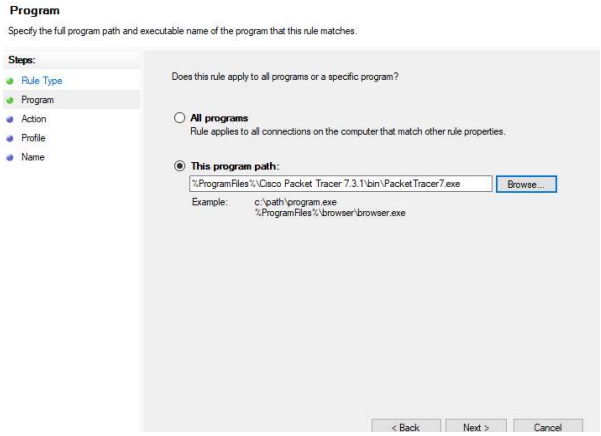


Step 6) Now Select program option and then click on next

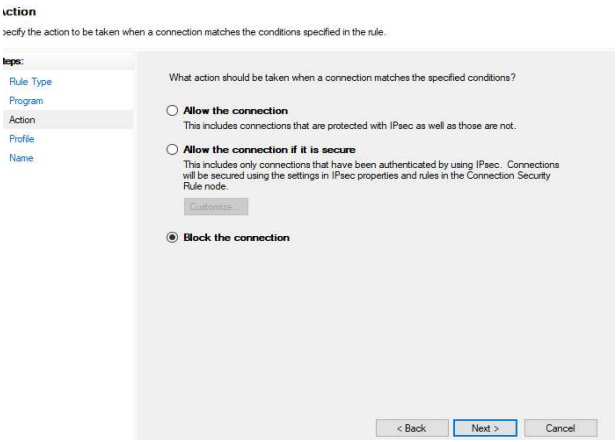


Bhumika Suhas Mane Roll No: 45

Step 7) Browse the program path of the file which is in C drive->Program Files->Cisco Packet Tracer 7.3.1->bin->PacketTracer.exe (You can choose any program path you want) and then click on next

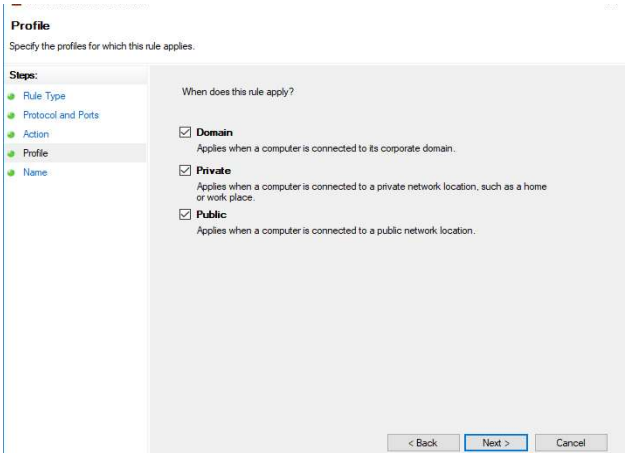


Step 8) Select Block the connection and click on next

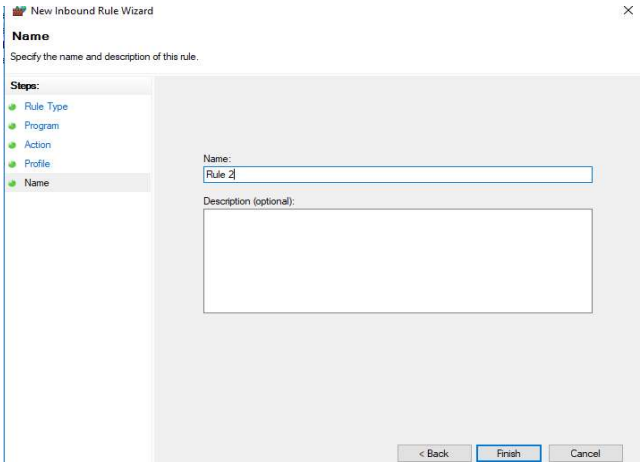


Step 9) Here, let the default changes be as it is and click on next

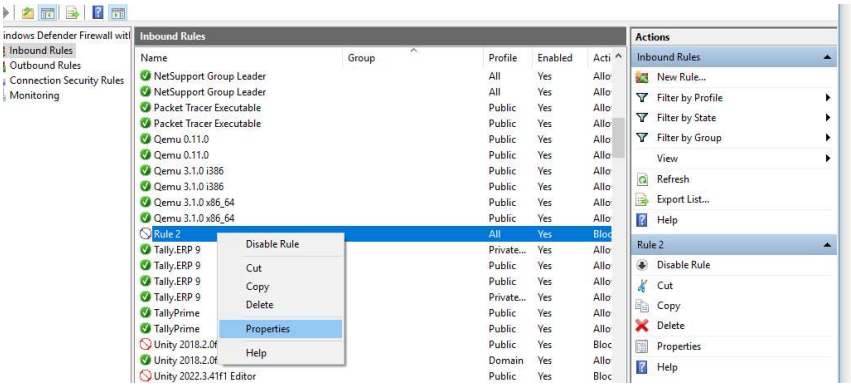
Bhumika Suhas Mane Roll No: 45



Step 10) Now type name as Rule 2 (the name by which we can identify our protocol) and then click on finish.

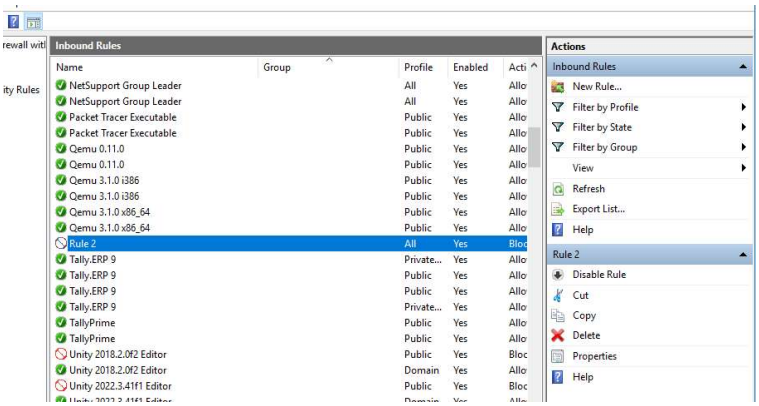


Step 11) Now right click on the Rule 2 and go to properties option. Don't change anything.



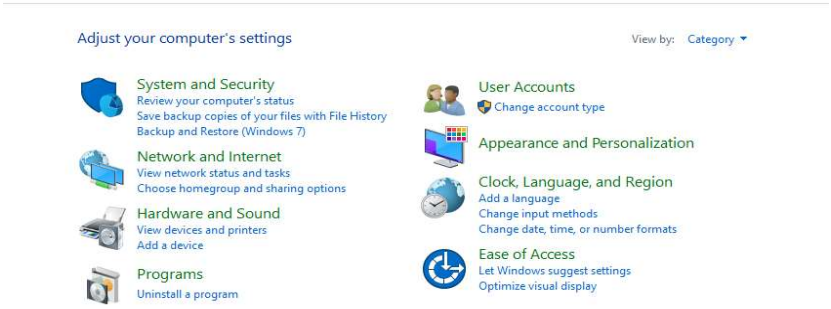
Bhumika Suhas Mane Roll No: 45

Step 12) We can also change the properties or disable the protocol using properties OR through the right side panel.



c) A Website

Step 1) Open control panel



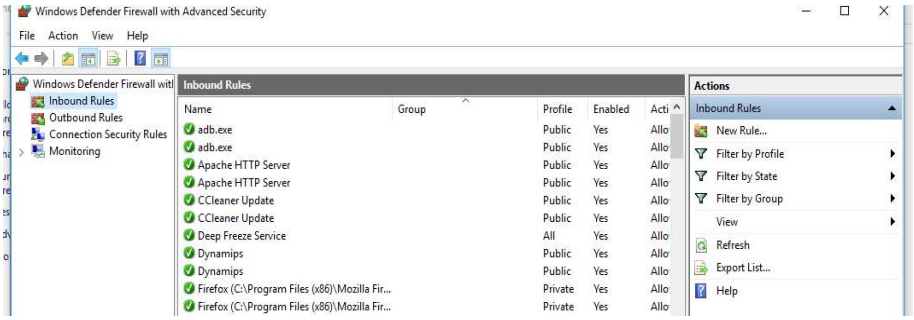
Step 2) Click on system and security

Step 3) Open Windows Defender Firewall

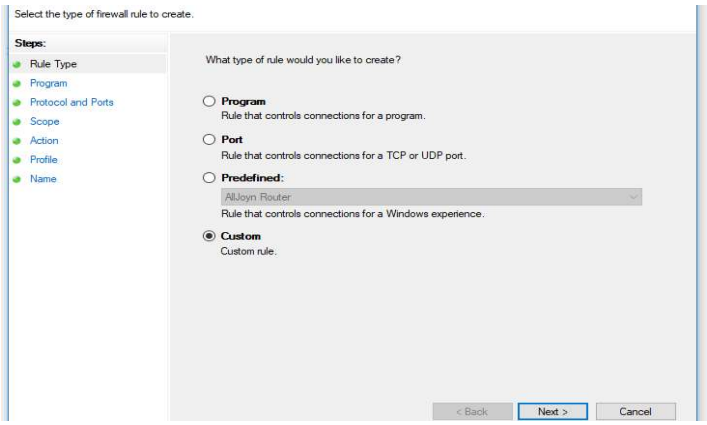
Bhumika Suhas Mane Roll No: 45

Step 4) Click on advanced settings

Step 5) Click on inbound rules and then on right side ->click on new rule.

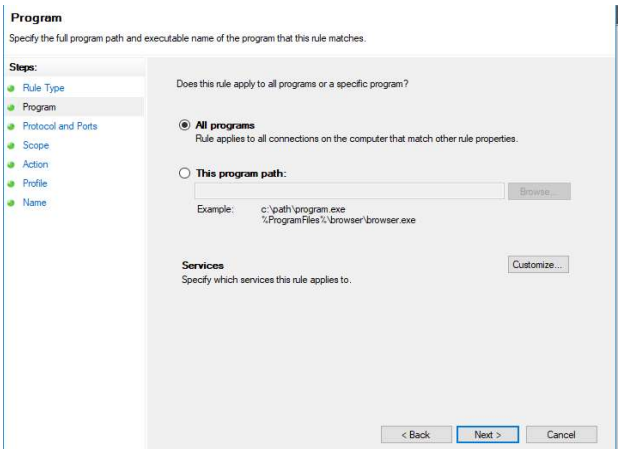


Step 6) Now Select custom option and then click on next.

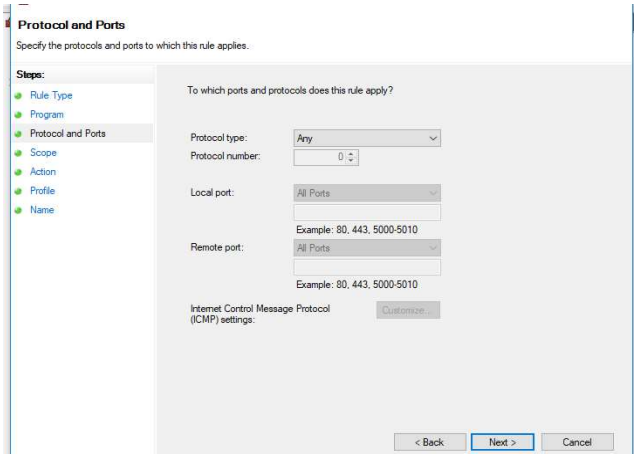


Bhumika Suhas Mane Roll No: 45

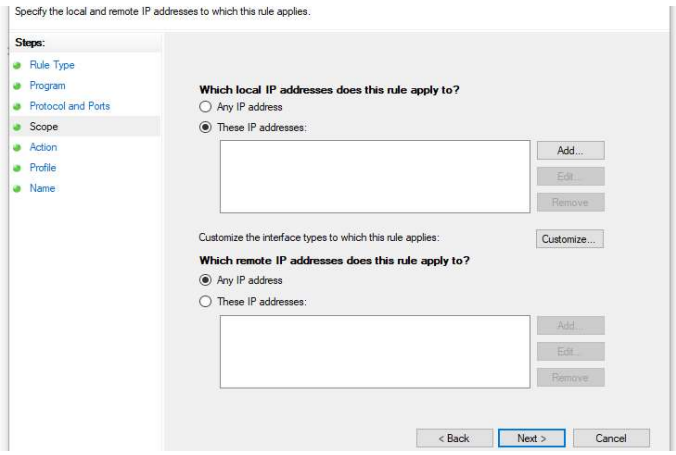
Step 7) Select the all programs option here and click next.



Step 8)Don't make any changes here, click on next.

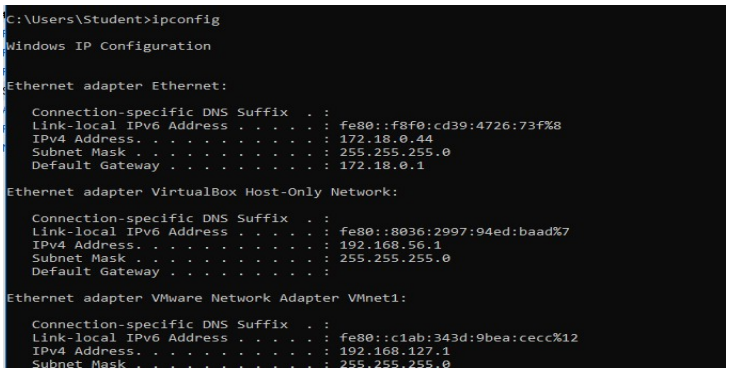


Step 9)Select these IP addresses for both of the options.



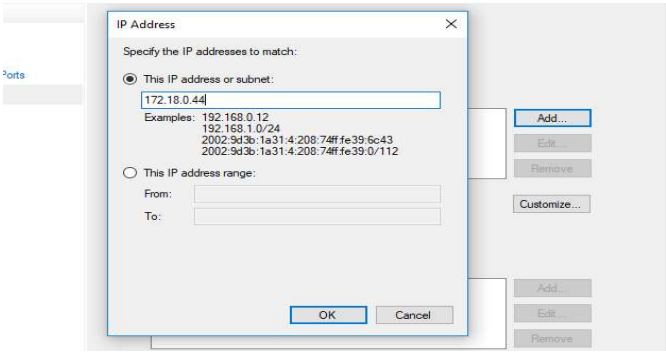
Bhumika Suhas Mane Roll No: 45

Step 10) Go to command prompt and check the IP address of your PC by using **ipconfig** command and copy it.

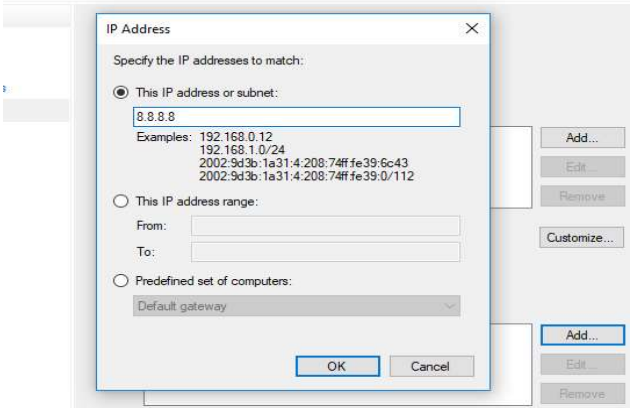


```
C:\Users\Student>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::f8f0:cd39:4726:73f%8
   IPv4 Address. . . . . . . . . . . : 172.18.0.44
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 172.18.0.1

Ethernet adapter VirtualBox Host-Only Network:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::8036:2997:94ed:baad%7
   IPv4 Address. . . . . . . . . . . : 192.168.56.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Ethernet adapter VMware Network Adapter VMnet1:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::c1ab:343d:9bea:cecc%12
   IPv4 Address. . . . . . . . . . . : 192.168.127.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
```

Step 11) Click on Add then paste the IP Address of your respective PCs in ‘This IP address or subnet’ and click on OK.

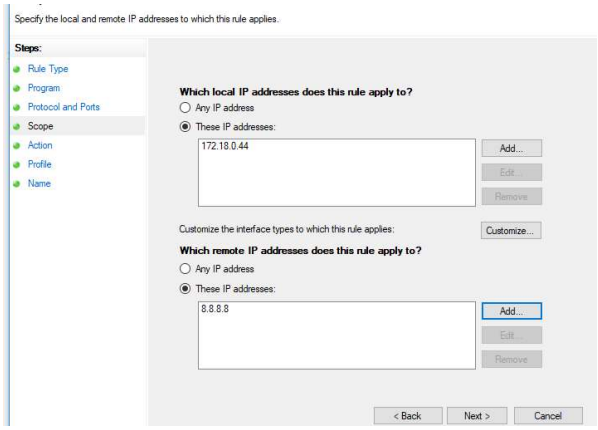


Step 12) Now on the remote IP address type any website’s IP address you want to block the connection of (8.8.8.8 is the IP Address of Google).

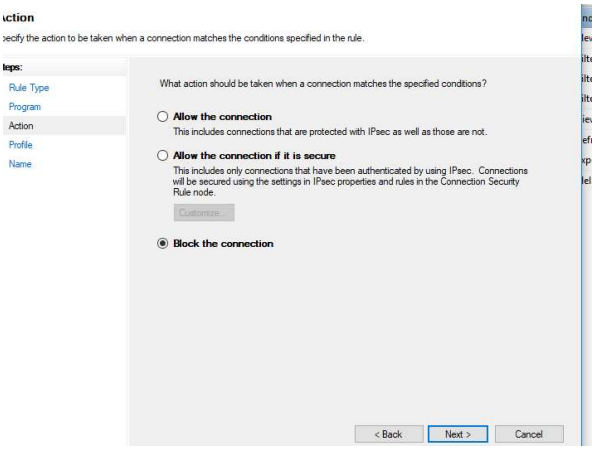


Bhumika Suhas Mane Roll No: 45

Step 13) The IP addresses which we have written will be shown as below.

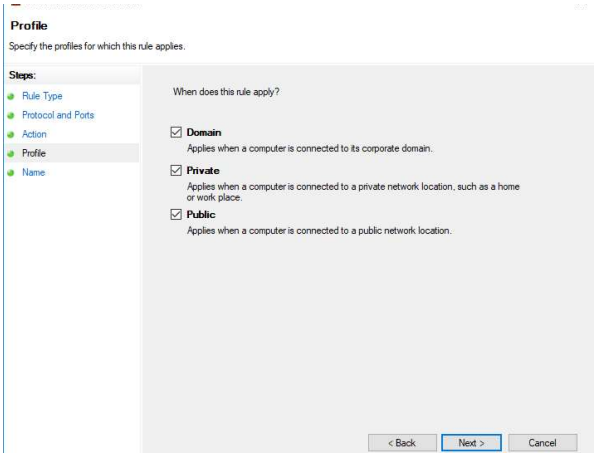


Step 14) Select Block the connection and click on next

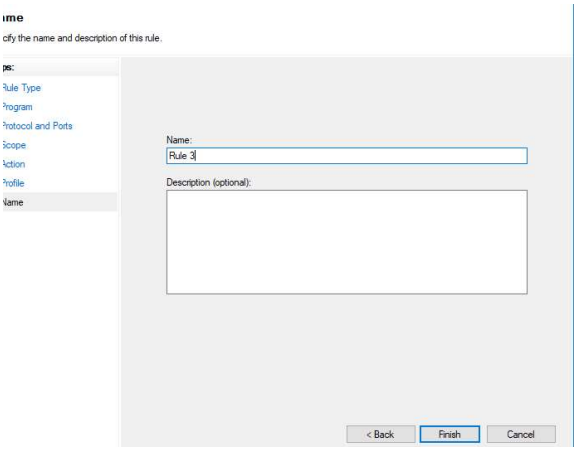


Step 15) Here, let the default changes be as it is and click on next

Bhumika Suhas Mane Roll No: 45



Step 16) Now type name as Rule 3 (the name by which we can identify our protocol) and then click on finish.



Step 17) We can change the properties or disable the protocol using properties OR through the right side panel.