Experiment No:01

Encryption and Decryption

Program in C:

```
#include <stdio.h>
int main()
{
  int i, x;
  char str[100];

  printf("\nPlease enter a
  string:\t");
  gets(str);

  printf("\nPlease choose following
  options:\n");
  printf("1 = Encrypt the string.\n");
  printf("2 = Decrypt  the
  string.\n");
  scanf("%d", &x);

  //using switch case
  statementsswitch(x)
  {
  case 1:
    for(i = 0; (i < 100 && str[i] != '\0'); i++)
      str[i] = str[i] + 3; //the key for encryption is 3 that is added to ASCII value

    printf("\nEncrypted
    string: %s\n", str);
    break;

  case 2:
    for(i = 0; (i < 100 && str[i] != '\0'); i++)
      str[i] = str[i] - 3; //the key for encryption is 3 that is subtracted to ASCII
      value

    printf("\nDecrypted
    string: %s\n", str);
    break;

  default:

    printf("\nEr
    ror\n");
    return 0;
     }
```

Output:

OnlineGDB beta
online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Jobs new
Sign Up
Login

214K

About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2023 GDB Online

Run | Debug | Stop | Share | Save | { } Beautify | Language C

main.c

```c
#include <stdio.h>
int main()
{
 int i, x;
 char str[100];
 printf("\nPlease enter a string:\t");
 gets(str);
 printf("\nPlease choose following options:\n");
 printf("1 = Encrypt the string.\n");
 printf("2 = Decrypt the string.\n");
 scanf("%d", &x);
 //using switch case statements
 switch(x)
 {
```

input

```
Please enter a string:  pratik

Please choose following options:
1 = Encrypt the string.
2 = Decrypt the string.
1

Encrypted string: sudwln


...Program finished with exit code 0
Press ENTER to exit console.
```

OnlineGDB beta
online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Jobs new
Sign Up
Login

214K

About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2023 GDB Online

Run | Debug | Stop | Share | Save | { } Beautify | Language C

main.c

```c
#include <stdio.h>
int main()
{
 int i, x;
 char str[100];
 printf("\nPlease enter a string:\t");
 gets(str);
 printf("\nPlease choose following options:\n");
 printf("1 = Encrypt the string.\n");
 printf("2 = Decrypt the string.\n");
 scanf("%d", &x);
 //using switch case statements
 switch(x)
 {
```

input

```
Please enter a string:  sudwln

Please choose following options:
1 = Encrypt the string.
2 = Decrypt the string.
2

Decrypted string: pratik


...Program finished with exit code 0
Press ENTER to exit console.
```

Experiment No. 2(a)

Enter Email ID & Password

Program in C++:

```
#include<iostream> usingnamespacestd;


classLoginManager
{
            public:
      stringuserNameAttempt;
      stringpassWordAttempt;

      LoginManager(){

       accessGranted=0;
      }

      voidlogin(){
         cout<<"Heyyouneedtoenteryourpasswordanduser.\nUsername:";
         cin>>userNameAttempt;
         if(userNameAttempt==userName){ cout<<"Password:";
           cin>>passWordAttempt;

           if(passWordAttempt==passWord){

              cout<<"Hey,that'sright.";

           }
        }
      }
   private: stringpassWord="gaikwad@123";
      stringuserName=
      "poonamgaikwad2002@email.com";
      boolaccessGranted;
};

intmain()
{
   LoginManagerloginManagerObj;
   loginManagerObj.login();
}
```

Output:

OnlineGDB beta
online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Jobs new
Sign Up
Login

Run | Debug | Stop | Share | Save | { } Beautify | Language C++

main.cpp

```cpp
#include <iostream>
using namespace std;
class LoginManager{
 public:
 string userNameAttempt;
 string passWordAttempt;
 LoginManager(){
 accessGranted = 0;
 }
 void login(){
 cout << "Hey you need to enter your password and user.\nUsername:";
 cin >> userNameAttempt;
 if(userNameAttempt==userName){
 cout << "Password:";
 cin >> passWordAttempt;
 if(passWordAttempt==passWord){
 cout << "Hey, that's right.";
 }
 }
```

input

```
Hey you need to enter your password and user.
Username:pratik@email.com
Password:pratik@123
Hey, that's right.

...Program finished with exit code 0
Press ENTER to exit console.
```

About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy

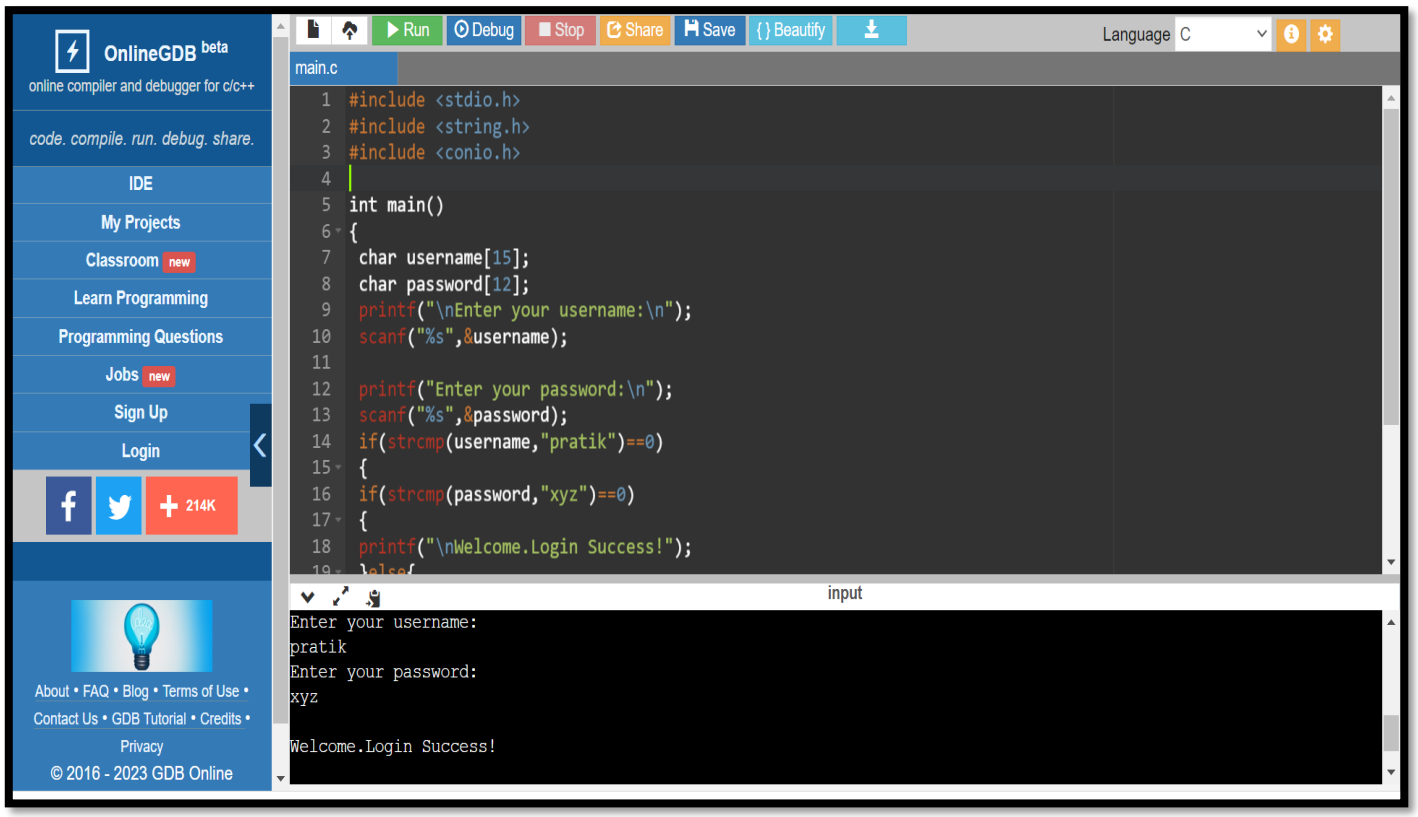
Experiment No. 2(b)

Enter Username & Password

Program in C:

```
#include<stdio.h>
 #include<string.h>
 #include<conio.h>

intmain()
{
   charusername[15];
   charpassword[12];


   printf("Enteryourusername:\n");
   scanf("%s",&username);

   printf("Enteryourpassword:\n");
   scanf("%s",&password);

   if(strcmp(username,"Poonam")==0){if(strcmp(password,"
      abcd")==0){

      printf("\nWelcome.LoginSuccess!");


      }else{ printf("\nwrongpassword");
} }else{
   printf("\nUserdoesn'texist");
}

   return0;

}
```

OUTPUT:


OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Jobs new
Sign Up
Login
214K
About • FAQ • Blog • Terms of Use •
Contact Us • GDB Tutorial • Credits •
Privacy
© 2016 - 2023 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language C
main.c
#include <stdio.h>
#include <string.h>
#include <conio.h>
int main()
{
char username[15];
char password[12];
printf("\nEnter your username:\n");
scanf("%s",&username);
printf("Enter your password:\n");
scanf("%s",&password);
if(strcmp(username,"pratik")==0)
{
if(strcmp(password,"xyz")==0)
{
printf("\nWelcome.Login Success!");
input
Enter your username:
pratik
Enter your password:
xyz
Welcome.Login Success!

Experiment No. 2(c)

Hiding Password

Program in C++:

```
#include <iostream>
#include <string>

using namespace std;
void askForSecretPassword(void){
	string password;
	cout<<"Enter your password: ";
	cin>>password;
	int len=password.length();
	system("cls");
	cout<<"Enter numeric password: ";
	for(int i=0;i<len;i++){
		cout<<"*";
	}cout<<endl;
};
int main(){

	askForSecretPassword();
	system("pause");
 return 0;
}
```

C:\Users\Shree\Documents\EXP2NS_A.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Cla EXP2NS_A.cpp EXP2NS_B.cpp

```cpp
#include <iostream>
#include <string>

using namespace std;
void askForSecretPassword(void)
{
    string password;
    cout<<"Enter your password: ";
    cin>>password;
    int len=password.length();
    system("cls");
    cout<<"Enter numeric password: ";
    for(int i=0;i<len;i++)
    {
      cout<<"*";
    }
    cout<<endl;
```

Output:

C:\Users\Shree\Documents\EXP2NS_A.exe

```
Enter numeric password: ********
Press any key to continue . . .
```

Experiment No. 3 (a)
Caesar Cipher

Program in C++:

```
// A C++ program to illustrate Caesar Cipher Technique
#include <iostream>
using namespace std;

   // This function receives text and shift and
  // returns the encrypted text
string encrypt(string text, int s)
{
        string result = "";

        // traverse text
        for (int i=0;i<text.length();i++)
        {
                // apply transformation to each character
                // Encrypt Uppercase letters
            if (isupper(text[i]))
                result += char(int(text[i]+s-65)%26 +65);

        // Encrypt Lowercase letters
        else
            result += char(int(text[i]+s-97)%26 +97);
        }

        // Return the resulting string
        return result;
}

// Driver program to test the above function
int main()
{
        string text="ATTACKATONCE";
        int s = 4;
        cout << "Text : " << text;
        cout << "\nShift: " << s;
        cout << "\nCipher: " << encrypt(text, s);
        return 0;
}
```

C:\Users\Shree\Documents\Exp3_NS_A.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Cla EXP2NS_A.cpp EXP2NS_B.cpp Exp3_NS_A.cpp

```cpp
// A C++ program to illustrate Caesar Cipher Technique
#include <iostream>
using namespace std;
// This function receives text and shift and
// returns the encrypted text
string encrypt(string text, int s)
{
string result = "";
// traverse text
for (int i=0;i<text.length();i++)
{
// apply transformation to each character
// Encrypt Uppercase letters
if (isupper(text[i]))
result += char(int(text[i]+s-65)%26 +65);
// Encrypt Lowercase letters
else
```

Output:

C:\Users\Shree\Documents\Exp3nsA.exe

```
Text : ATTACKATONCE
Shift: 4
Cipher: EXXEGOEXSRGI
--------------------------------
Process exited after 1.104 seconds with return value 0
Press any key to continue . . .
```

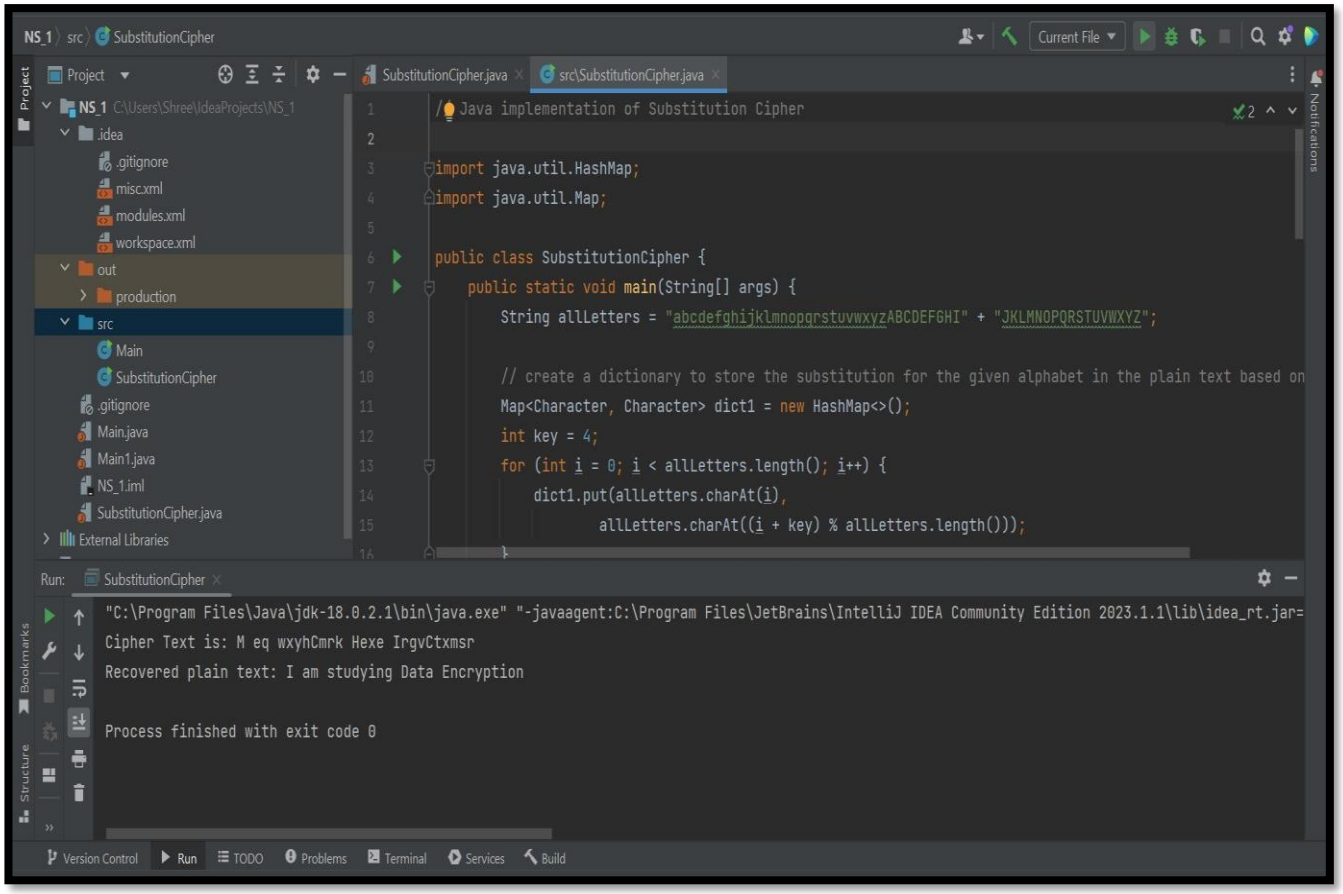
Experiment No. 3 (b)

Substitution Technique

```
// Java implementation of Substitution Cipher

import java.io.*;

import java.util.*;

import java.util.HashMap;

import java.util.Map;

public class SubstitutionCipher{

	public static void main(String[] args) {

		String allLetters = "abcdefghijklmnopqrstuvwxyzABCDEFGHI" +
"JKLMNOPQRSTUVWXYZ";

// create a dictionary to store the substitution for the given alphabet in the plain //text based on the key

		Map<Character, Character> dict1 = new HashMap<>();

		int key = 4;

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

			dict1.put(allLetters.charAt(i),

				allLetters.charAt((i + key) % allLetters.length()));

		}


		String plainText = "I am studying Data Encryption";

		StringBuilder cipherText = new StringBuilder();


		// loop to generate ciphertext

		for (char c : plainText.toCharArray()) {

			if (allLetters.indexOf(c) != -1) {

				cipherText.append(dict1.get(c));

			} else {

				cipherText.append(c);
```

```
                }
        }

        System.out.println("Cipher Text is: " + cipherText);

        // create a map to store the substitution for the given alphabet in the cipher text based
on the key
        Map<Character, Character> dict2 = new HashMap<>();
        for (int i = 0; i < allLetters.length(); i++) {
                dict2.put(allLetters.charAt(i),
                                allLetters.charAt((i - key + allLetters.length()) %
allLetters.length()));
        }

        StringBuilder decryptedText = new StringBuilder();

        // loop to recover plain text
        for (char c : cipherText.toString().toCharArray()) {
                if (allLetters.indexOf(c) != -1) {
                        decryptedText.append(dict2.get(c));
                } else {
                        decryptedText.append(c);
                }
        }

        System.out.println("Recovered plain text: " + decryptedText);
    }
}
```

OUTPUT:


NS_1 > src > SubstitutionCipher
Current File
Project
NS_1 C:\Users\Shree\IdeaProjects\NS_1
.idea
.gitignore
misc.xml
modules.xml
workspace.xml
out
production
src
Main
SubstitutionCipher
.gitignore
Main.java
Main1.java
NS_1.iml
SubstitutionCipher.java
External Libraries
SubstitutionCipher.java
src\SubstitutionCipher.java
/Java implementation of Substitution Cipher
import java.util.HashMap;
import java.util.Map;
public class SubstitutionCipher {
public static void main(String[] args) {
String allLetters = "abcdefghijklmnopqrstuvwxyzABCDEFGHI" + "JKLMNOPQRSTUVWXYZ";
// create a dictionary to store the substitution for the given alphabet in the plain text based on
Map<Character, Character> dict1 = new HashMap<>();
int key = 4;
for (int i = 0; i < allLetters.length(); i++) {
dict1.put(allLetters.charAt(i),
allLetters.charAt((i + key) % allLetters.length()));
Run: SubstitutionCipher
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.1.1\lib\idea_rt.jar=
Cipher Text is: M eq wxyhCmrk Hexe IrgvCtxmsr
Recovered plain text: I am studying Data Encryption
Process finished with exit code 0
Version Control
Run
TODO
Problems
Terminal
Services
Build

Experiment No. 5

Symmetric and Asymmetric Cryptography using RSA Algorithm

Program in java:

Asymmetric cryptography algorithm

Program of RSA Alogorithm

Program in C: Program of RSA Alogorithm

```
#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#include<string.h>

long int p,q,n,t,flag,e[100],d[100],temp[100],j,m[100],en[100],i;

char msg[100];

int prime(long int);

void ce();

long int cd(long int);

void encrypt();

void decrypt();

int main()

{

 printf("\nENTER FIRST PRIME NUMBER\n");

 scanf("%ld",&p);

 flag=prime(p);

 if(flag==0)

 {

 printf("\nWRONG INPUT\n");

 exit(1);

 }

 printf("\nENTER ANOTHER PRIME NUMBER\n");

 scanf("%ld",&q);
```

```
 flag=prime(q);
 if(flag==0||p==q)
 {
 printf("\nWRONG INPUT\n");
 exit(1);
 }
 printf("\nENTER MESSAGE\n");
 fflush(stdin);
 scanf("%s",msg);
 for(i=0;msg[i]!=NULL;i++)
 m[i]=msg[i];
 n=p*q;
 t=(p-1)*(q-1);
 ce();
 printf("\nPOSSIBLE VALUES OF e AND d ARE\n");
 for(i=0;i<j-1;i++)
 printf("\n%ld\t%ld",e[i],d[i]);
 encrypt();
 decrypt();
 return 0;
}
int prime(long int pr)
{
 int i;
 j=sqrt(pr);
 for(i=2;i<=j;i++)
 {
 if(pr%i==0)
 return 0;
 }
 return 1;
```

```
}
void ce()
{
 int k;
 k=0;
 for(i=2;i<t;i++)
 {
 if(t%i==0)
 continue;
 flag=prime(i);
 if(flag==1&&i!=p&&i!=q)
 {
 e[k]=i; flag=cd(e[k]);
 if(flag>0)
 {
 d[k]=flag;
 k++;
 }
 if(k==99)
 break;
 }
 }
}
long int cd(long int x)
{
 long int k=1;
 while(1)
 {
 k=k+t;
 if(k%x==0)
 return(k/x);
```

```
 }
}
void encrypt()
{
 long int pt,ct,key=e[0],k,len;
 i=0;
 len=strlen(msg);
 while(i!=len)
 {
 pt=m[i];
pt=pt-96;
k=1;
 for(j=0;j<key;j++)
 {
 k=k*pt;
 k=k%n;
 }
 temp[i]=k;
 ct=k+96;
 en[i]=ct;
 i++;
 }
 en[i]=-1;
 printf("\nTHE ENCRYPTED MESSAGE IS\n");
 for(i=0;en[i]!=-1;i++)
 printf("%c",en[i]);
}
void decrypt()
{
 long int pt,ct,key=d[0],k;
 i=0;
```

```
while(en[i]!=-1)
{
ct=temp[i];
k=1;
for(j=0;j<key;j++)
{
k=k*ct;
k=k%n;
}
pt=k+96;
m[i]=pt;
i++;
}
m[i]=-1;
printf("\nTHE DECRYPTED MESSAGE IS\n");
for(i=0;m[i]!=-1;i++)
printf("%c",m[i]);
}
```

D:\3133 NS\exp5RSA.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

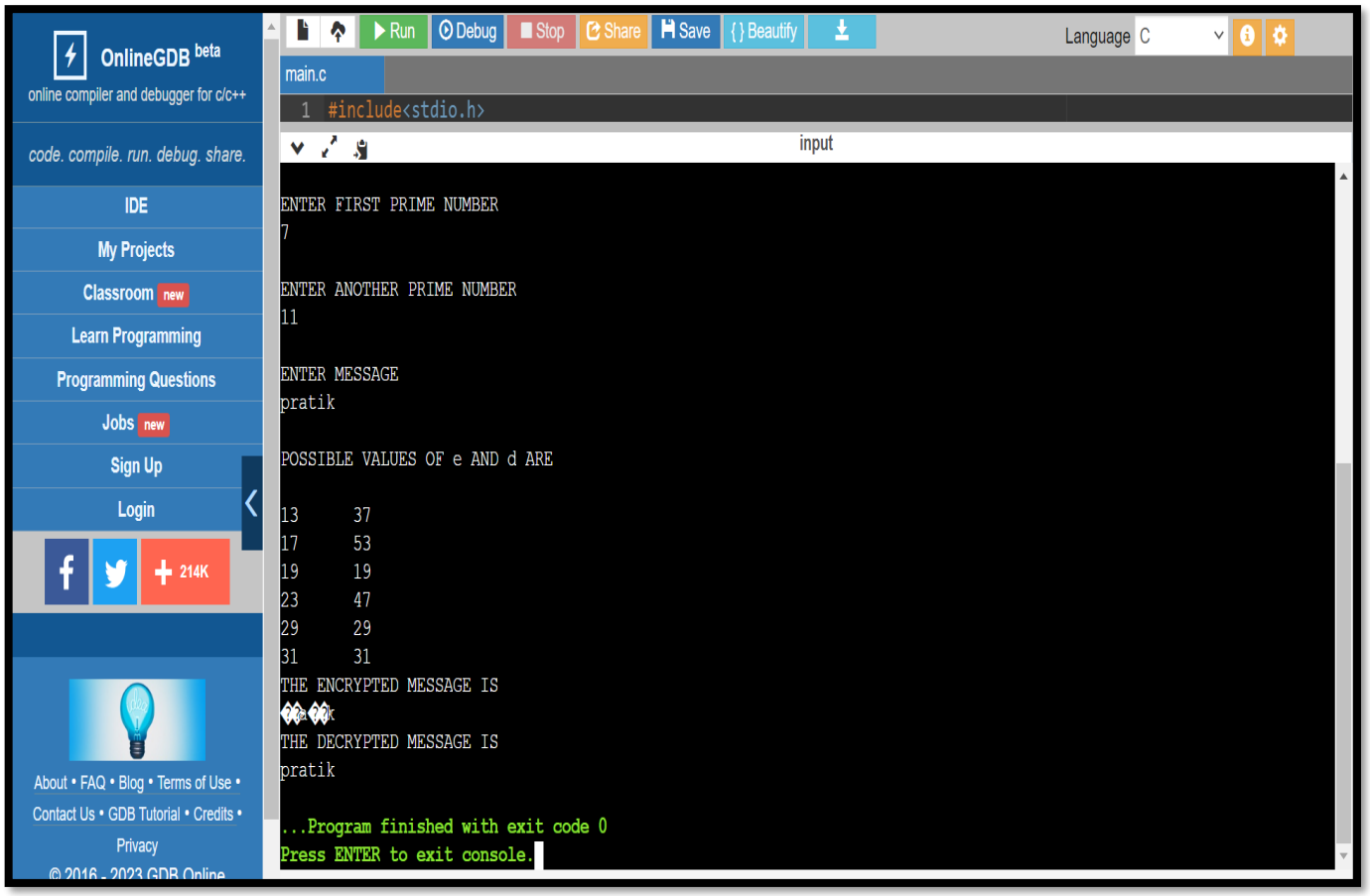
TDM-GCC 4.9.2 64-bit Release

(globals)

exp5RSA.cpp

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<math.h>
4  #include<string.h>
5  long int p,q,n,t,flag,e[100],d[100],temp[100],j,m[100],en[100],i;
6  char msg[100];
7  int prime(long int);
8  void ce();
9  long int cd(long int);
10 void encrypt();
11 void decrypt();
12 int main()
13 {
14  printf("\nENTER FIRST PRIME NUMBER\n");
15  scanf("%ld",&p);
16  flag=prime(p);
17  if(flag==0)
18  {
19  printf("\nWRONG INPUT\n");
20  exit(1);
21  }
22  printf("\nENTER ANOTHER PRIME NUMBER\n");
23  scanf("%ld",&q);
24  flag=prime(q);
25  if(flag==0||p==q)
26  {
```

OUTPUT:-

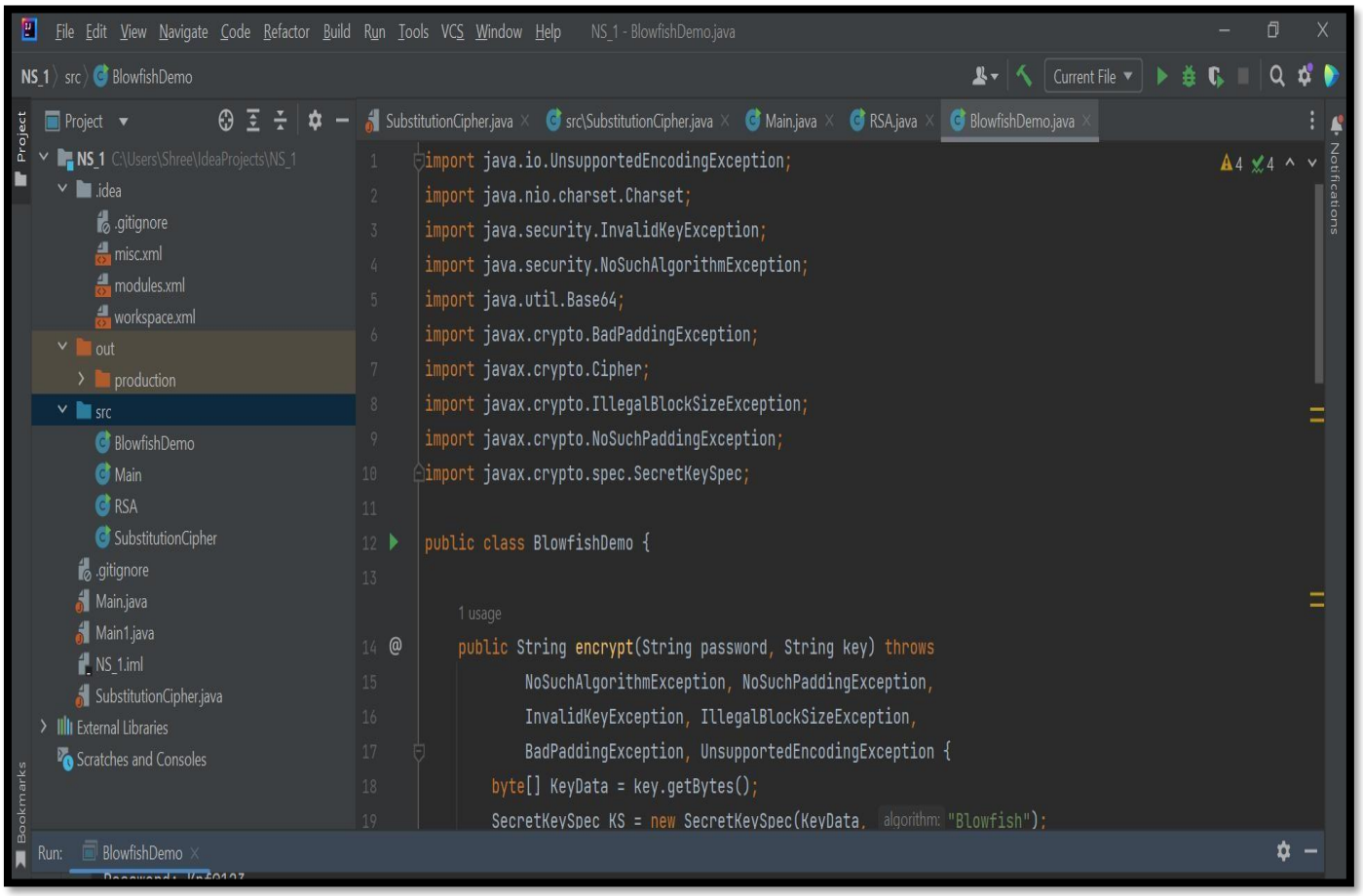

OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Jobs new
Sign Up
Login
214K
About • FAQ • Blog • Terms of Use •
Contact Us • GDB Tutorial • Credits •
Privacy
© 2016 - 2023 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language C
main.c
1 #include<stdio.h>
input
ENTER FIRST PRIME NUMBER
7
ENTER ANOTHER PRIME NUMBER
11
ENTER MESSAGE
pratik
POSSIBLE VALUES OF e AND d ARE
13 37
17 53
19 19
23 47
29 29
31 31
THE ENCRYPTED MESSAGE IS
THE DECRYPTED MESSAGE IS
pratik
...Program finished with exit code 0
Press ENTER to exit console.

symmetric cryptography algorithm

Program of Blowfish Alogorithm

```
import java.io.UnsupportedEncodingException;
import java.nio.charset.Charset;
import  java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.util.Base64;
import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.spec.SecretKeySpec;
public class BlowfishDemo {

   public String encrypt(String password, String key) throws
            NoSuchAlgorithmException, NoSuchPaddingException,
            InvalidKeyException, IllegalBlockSizeException,
            BadPaddingException, UnsupportedEncodingException {
      byte[] KeyData = key.getBytes();
      SecretKeySpec KS = new SecretKeySpec(KeyData, "Blowfish");
      Cipher cipher = Cipher.getInstance("Blowfish");
      cipher.init(Cipher.ENCRYPT_MODE, KS);
      String encryptedtext = Base64.getEncoder().
         encodeToString(cipher.doFinal(password.getBytes("UTF-8")));
      return encryptedtext;

   }

   public String decrypt(String encryptedtext, String key)
       throws NoSuchAlgorithmException, NoSuchPaddingException,
         InvalidKeyException, IllegalBlockSizeException,
                BadPaddingException {
      byte[] KeyData = key.getBytes();
      SecretKeySpec KS = new SecretKeySpec(KeyData, "Blowfish");
      byte[] ecryptedtexttobytes = Base64.getDecoder().
                 decode(encryptedtext);
      Cipher cipher = Cipher.getInstance("Blowfish");
      cipher.init(Cipher.DECRYPT_MODE, KS);
      byte[] decrypted = cipher.doFinal(ecryptedtexttobytes);
      String decryptedString =
             new String(decrypted, Charset.forName("UTF-8"));
      return decryptedString;

   }

   public static void main(String[] args) throws Exception {
```

```
        final String password = "Knf@123";
        final String key = "knowledgefactory";
        System.out.println("Password: " + password);
        BlowfishDemo obj = new BlowfishDemo();
        String enc_output = obj.encrypt(password, key);
        System.out.println("Encrypted text: " + enc_output);
        String dec_output = obj.decrypt(enc_output, key);
        System.out.println("Decrypted text: " + dec_output);
    }
}
```



OUTPUT:-

```
Run: BlowfishDemo
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.1.1\lib\idea_rt.jar=
Password: Knf@123
Encrypted text: 4DTHqnctCuk=
Decrypted text: Knf@123

Process finished with exit code 0
```

Experiment No. 6

Implementation of DES Encryption

Program in Java:

```
import  javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import java.util.Base64;
class DESExample {
Cipher ecipher;
Cipher dcipher;
DESExample(SecretKey key) throws Exception {
        ecipher = Cipher.getInstance("DES");
        dcipher = Cipher.getInstance("DES");
        ecipher.init(Cipher.ENCRYPT_MODE, key);
        dcipher.init(Cipher.DECRYPT_MODE, key);
    }
       public String encrypt(String str) throws Exception {
        // Encode the string into bytes using utf-8
        byte[] utf8 = str.getBytes("UTF8");
         // Encrypt
        byte[] enc = ecipher.doFinal(utf8);
         // Encode bytes to base64 to get a string
         return Base64.getEncoder().encodeToString(enc);
         }
          public String decrypt(String str) throws Exception {
```

```
    // Decode base64 to get bytes
    byte[] dec = Base64.getDecoder().decode(str);


    byte[] utf8 = dcipher.doFinal(dec);
      // Decode using utf-8
    return new String(utf8, "UTF8");
  }
  public static void main(String[] argv) throws Exception {
    final String secretText = "Network Security";
    System.out.println("SecretText: " + secretText);
    SecretKey key = KeyGenerator.getInstance("DES").generateKey();
    DESExample encrypter = new DESExample(key);
    String encrypted = encrypter.encrypt(secretText);
    System.out.println("Encrypted Value: " + encrypted);
    String decrypted = encrypter.decrypt(encrypted);
    System.out.println("Decrypted: " + decrypted);
    }
}
```

```java
import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import java.util.Base64;

class DESExample {

    3 usages
    Cipher ecipher;
    3 usages
    Cipher dcipher;

    1 usage
    DESExample(SecretKey key) throws Exception {
        ecipher = Cipher.getInstance( transformation: "DES");
        dcipher = Cipher.getInstance( transformation: "DES");
        ecipher.init(Cipher.ENCRYPT_MODE, key);
        dcipher.init(Cipher.DECRYPT_MODE, key);
    }
```

OUTPUT :-

```
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community
SecretText: Network Security
Encrypted Value: nTNAqrOOOmGXLlokgNMDOMorjp5EHRus
Decrypted: Network Security

Process finished with exit code 0
```

Experiment No. 7

Implementation of AES

```
import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.crypto.spec.GCMParameterSpec;
import java.util.Base64;
public class AES_ENCRYPTION {
   private SecretKey key;
   private final int KEY_SIZE = 128;
   private final int DATA_LENGTH = 128;
   private Cipher encryptionCipher;

   public void init() throws Exception {
      KeyGenerator keyGenerator = KeyGenerator.getInstance("AES");
      keyGenerator.init(KEY_SIZE);
      key = keyGenerator.generateKey();
   }
   public String encrypt(String data) throws Exception {
      byte[] dataInBytes = data.getBytes();
      encryptionCipher = Cipher.getInstance("AES/GCM/NoPadding");
      encryptionCipher.init(Cipher.ENCRYPT_MODE, key);
      byte[] encryptedBytes = encryptionCipher.doFinal(dataInBytes);
      return encode(encryptedBytes);
   }
   public String decrypt(String encryptedData) throws Exception {
      byte[] dataInBytes = decode(encryptedData);
      Cipher decryptionCipher = Cipher.getInstance("AES/GCM/NoPadding");
      GCMParameterSpec spec = new GCMParameterSpec(DATA_LENGTH, encryptionCipher.getIV());
```

```
        decryptionCipher.init(Cipher.DECRYPT_MODE, key, spec);
        byte[] decryptedBytes = decryptionCipher.doFinal(dataInBytes);
        return new String(decryptedBytes);
    }
    private String encode(byte[] data) {
        return Base64.getEncoder().encodeToString(data);
    }

    private byte[] decode(String data) {
        return Base64.getDecoder().decode(data);
    }
    public static void main(String[] args) {
        try {
            AES_ENCRYPTION aes_encryption = new AES_ENCRYPTION();
            aes_encryption.init();
            String encryptedData = aes_encryption.encrypt("Hello, welcome to the encryption world");
            String decryptedData = aes_encryption.decrypt(encryptedData);

            System.out.println("Encrypted Data : " + encryptedData);
            System.out.println("Decrypted Data : " + decryptedData);
        } catch (Exception ignored) {
        }
    }
}
```

```java
import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.crypto.spec.GCMParameterSpec;
import java.util.Base64;
public class AES_ENCRYPTION {
    private SecretKey key;
    private final int KEY_SIZE = 128;
    private final int DATA_LENGTH = 128;
    private Cipher encryptionCipher;

    public void init() throws Exception {
        KeyGenerator keyGenerator = KeyGenerator.getInstance( algorithm: "AES");
        keyGenerator.init(KEY_SIZE);
```

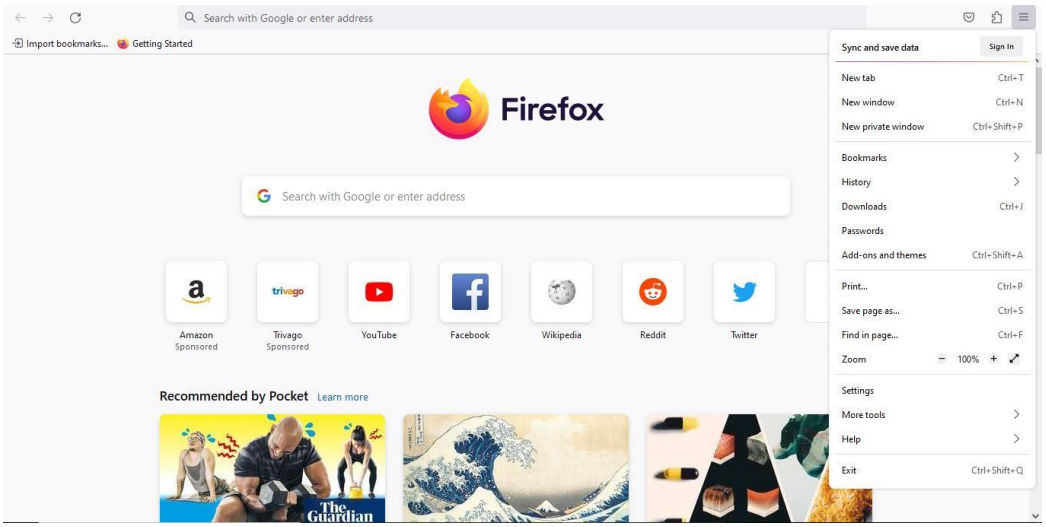
OUTPUT:-

```
Run: AES_ENCRYPTION
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\Intell
Encrypted Data : LiSW4RcyRLl/CBQApTeUt02vIUheauXI2tfV6qbnkAVHSgx4Npi6t+xRexf+GyBned5DsOet
Decrypted Data : Hello, welcome to the encryption world

Process finished with exit code 0
```

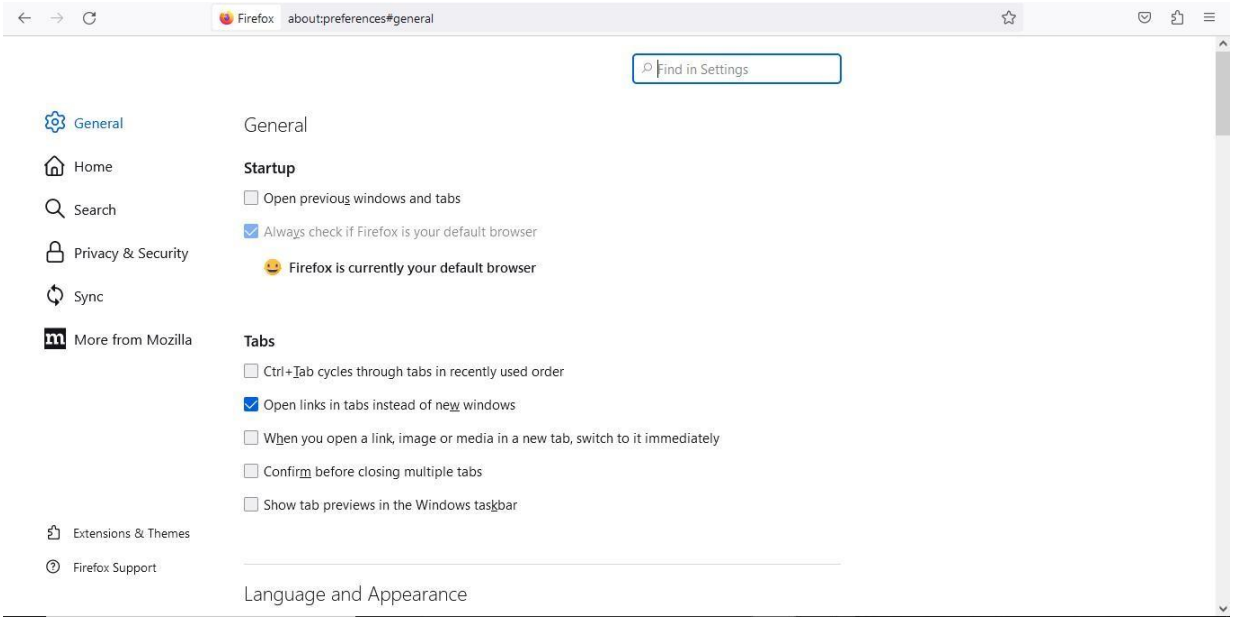
Experiment No. 8

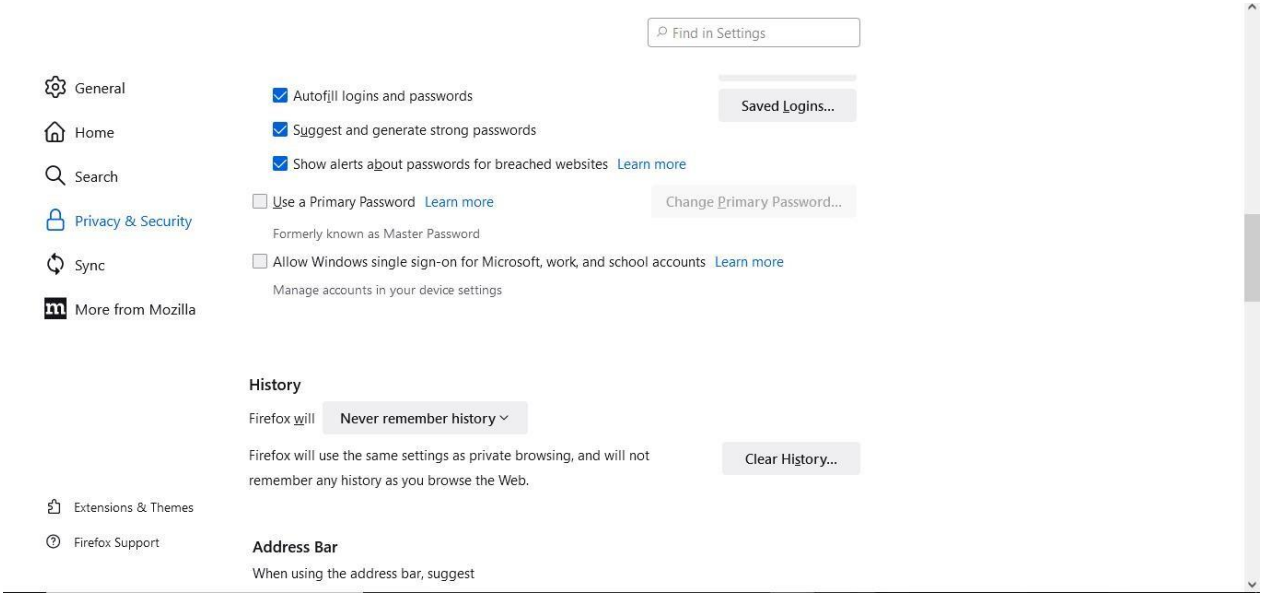
Steps to ensure security of web browser

1.Top Right Corner Go to the Options:

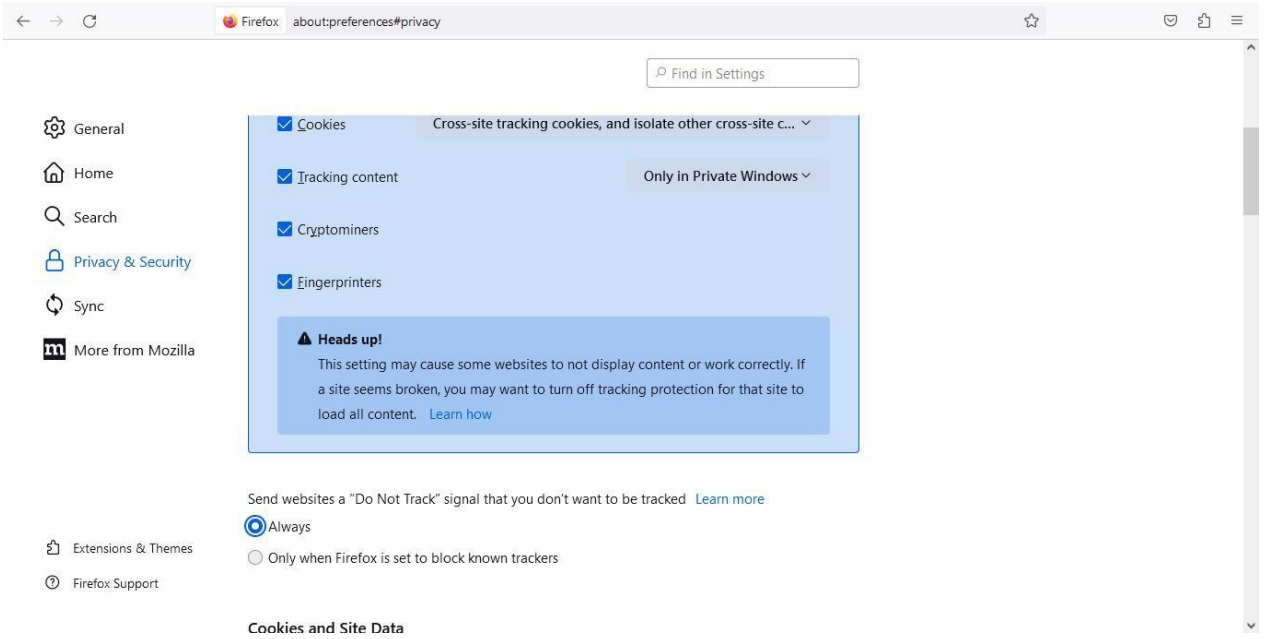


2. General ----→ History --- → Click on never remember History

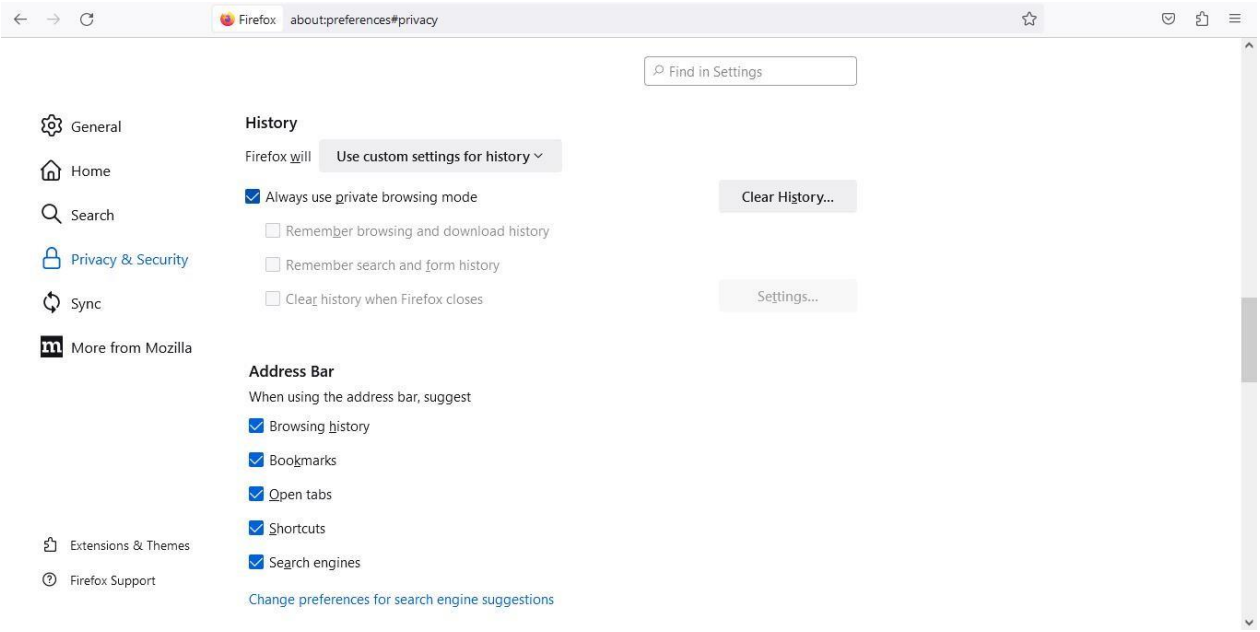


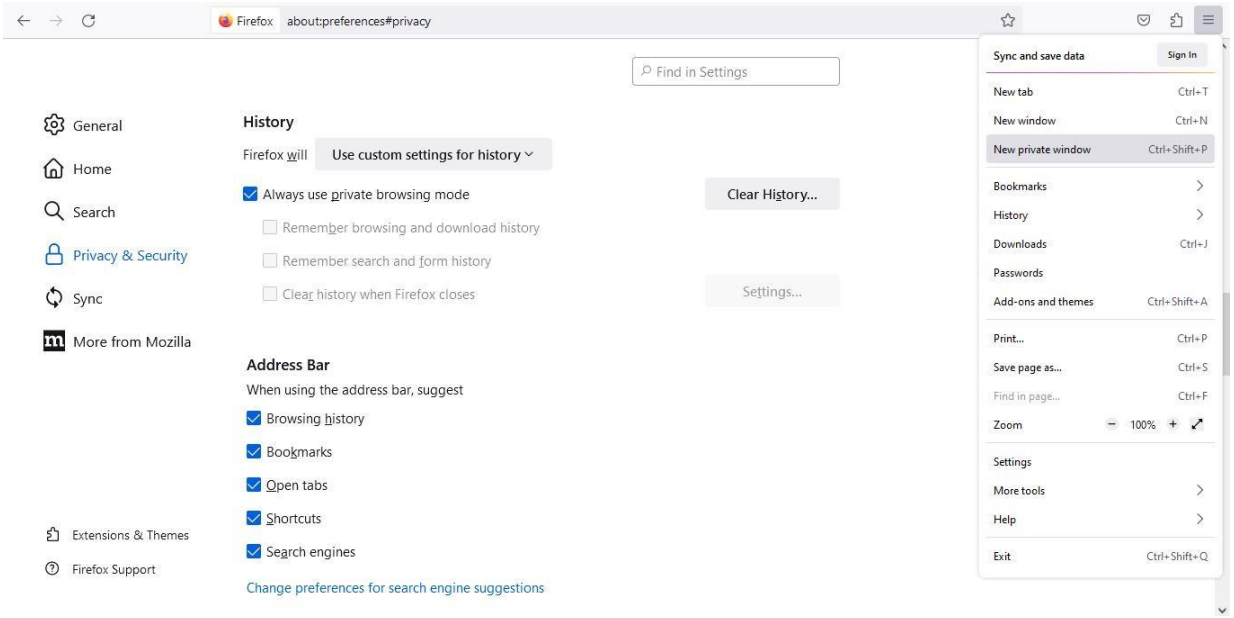


3.Privacy tab--->Tracking---->Manage your do not track setting--->Always Apply do not track:



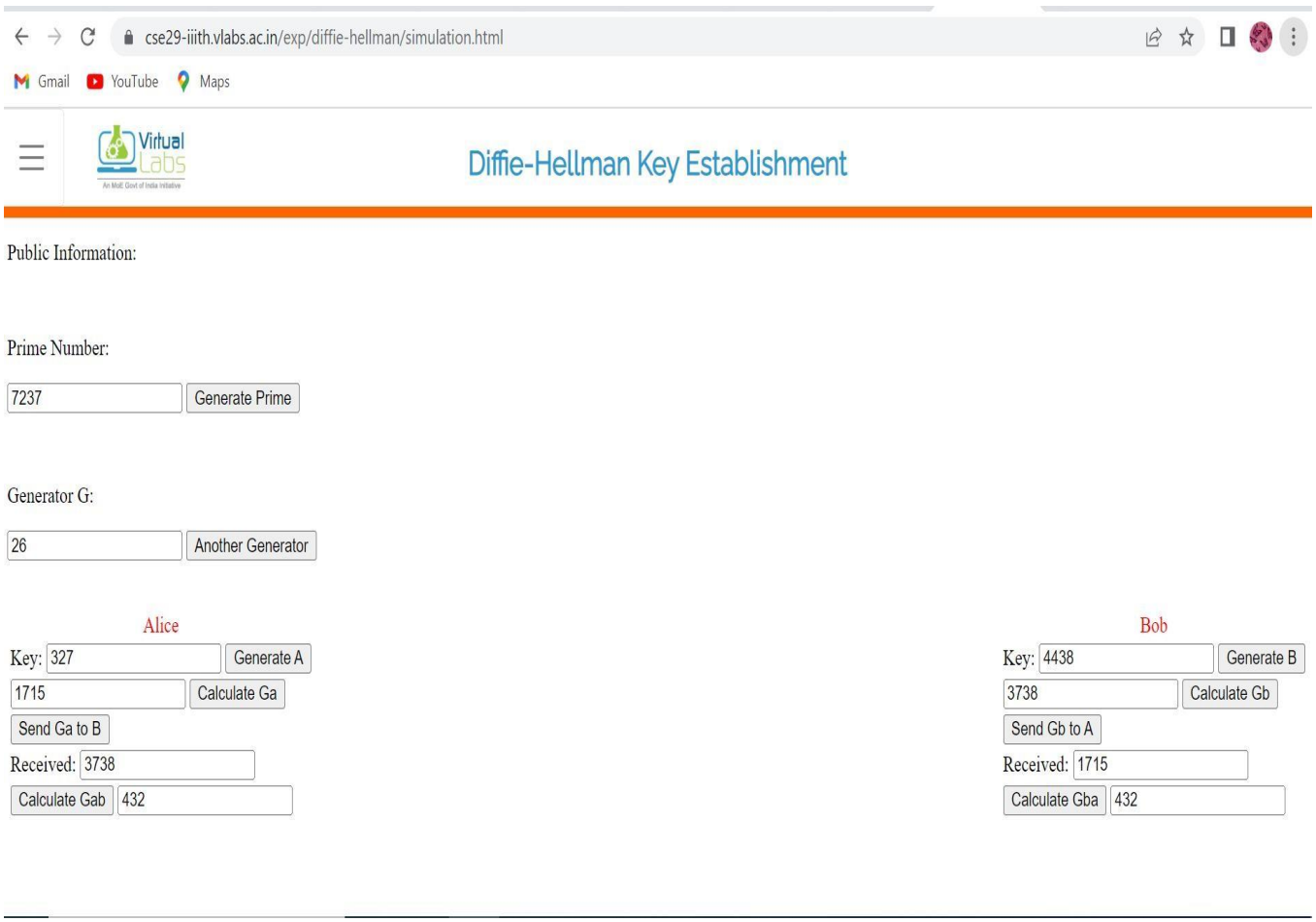
4. Privacy--->Tracking--->Use tracking Protection in private window--->Use custom setting for history--->Always use private browsing mode:


Firefox
about:preferences#privacy
Find in Settings
General
Home
Search
Privacy & Security
Sync
More from Mozilla
History
Firefox will
Use custom settings for history
Always use private browsing mode
Clear History...
Remember browsing and download history
Remember search and form history
Clear history when Firefox closes
Settings...
Address Bar
When using the address bar, suggest
Browsing history
Bookmarks
Open tabs
Shortcuts
Search engines
Change preferences for search engine suggestions
Extensions & Themes
Firefox Support


Firefox
about:preferences#privacy
Find in Settings
General
Home
Search
Privacy & Security
Sync
More from Mozilla
History
Firefox will
Use custom settings for history
Always use private browsing mode
Clear History...
Remember browsing and download history
Remember search and form history
Clear history when Firefox closes
Settings...
Address Bar
When using the address bar, suggest
Browsing history
Bookmarks
Open tabs
Shortcuts
Search engines
Change preferences for search engine suggestions
Extensions & Themes
Firefox Support
Sync and save data
Sign In
New tab Ctrl+T
New window Ctrl+N
New private window Ctrl+Shift+P
Bookmarks
History
Downloads Ctrl+J
Passwords
Add-ons and themes Ctrl+Shift+A
Print... Ctrl+P
Save page as... Ctrl+S
Find in page... Ctrl+F
Zoom 100%
Settings
More tools
Help
Exit Ctrl+Shift+Q

Experiment No. 9

DH Algorithm



Experiment No. 10

Digital Signatures

Virtual Labs

An MoE Govt of India Initiative

Digital Signatures Scheme

Digitally sign the plaintext with Hashed RSA.

Plaintext (string):

test | SHA-1

Hash output(hex):

a94a8fe5ccb19ba61c4c0873d391e987982fbbd3

Input to RSA(hex):

a94a8fe5ccb19ba61c4c0873d391e987982fbbd3 | Apply RSA

Digital Signature(hex):

```
2137c96309c7937b0f373da7f6ef3692a906038116af780df52674562132e26f
6d5545e1c774400736b645f8269facbaaf4c2aa54ebde5f1a563f651a3aaf522
fd4440b79a8107b31e890121745fbfea849a8291eea381a8e8d34bdaec706e5f
2ce0523dc38108558184f028be63c5b9077c6108f54951678b41054a860ace2e
```

Virtual Labs

An MoE Govt of India Initiative

Digital Signatures Scheme

Digital Signature(hex):

```
2137c96309c7937b0f373da7f6ef3692a906038116af780df52674562132e26f
6d5545e1c774400736b645f8269facbaaf4c2aa54ebde5f1a563f651a3aaf522
fd4440b79a8107b31e890121745fbfea849a8291eea381a8e8d34bdaec706e5f
2ce0523dc38108558184f028be63c5b9077c6108f54951678b41054a860ace2e
```

Digital Signature(base64):

```
ITfJYwnHk3sPNz2n9u82kqkGA4EWr3gN9SZ0ViEy4m9tVUXhx3RABza2Rfgmn6y6
r0wqpU695fGlY/ZRo6r1Iv1EQLeagQezHokBIXRfv+qEmoKR7qOBqOjTS9rscG5f
LOBSPcOBCFWBhPAovmPFuQd8YQj1SVFni0EFSoYKzi4=
```

Status:

Time: 14ms

RSA public key

Public exponent (hex, F4=0x10001):

10001

Modulus (hex):

```
a5261939975948bb7a58dffe5ff54e65f0498f9175f5a09288810b8975871e99
af3b5dd94057b0fc07535f5f97444504fa35169d461d0d30cf0192e307727c06
5168c788771c561a9400fb49175e9e6aa4e23fe11af69e9412dd23b0cb6684c4
c2429bce139e848ab26d0829073351f4acd36074eafd036a5eb83359d2a698d3
```

1024 bit | 1024 bit (e=3) | 512 bit | 512 bit (e=3)