Janvi Bhalala

Information & Network Security (INS)– 2160711

Lab Manual

Semester 7th

Academic Year 2020-21 Odd

VVP Engineering College,
Rajkot

Gujarat Technological
University

Table of Contents

Practical 1

BREAKING THE SHIFT CIPHER

Step To Breaking the shift cipher

STEP 1 : For the given ciphertext in the **PART I** of the simulation page, the first step is to decrypt it using each of the twenty-six different keys, k=0,1,...,25 and obtain the corresponding plaintexts. For decryption, you may use the tool given in the **PART III** of the simulation page.

STEP 2 : After each decryption, you may cut-and-paste the resultant plaintext in the scratch-pad in the **(PART II)** of the simulation page, if you need to remember it.

STEP 3 : Finally, observe the plaintexts and choose the most appropriate one (the one that is a meaningful English text) as the recovered plaintext and cut-and-paste it in the text-field named **PART IV** "Solution Plaintext". Also select the corresponding key in the text-field named "Key" and click on "Check My answer" Button.

STEP 4 [OPTIONAL] : Verify that your answer is correct, by encrypting the solution plaintext with your key.

Example:

Given a cipher text, find out the corresponding plain text using brute force attack.

Ciphertext: HAAHJR HA KHDU

For k=0

Cipher text: HAAHJR HA KHDU
plain text: haahjr ha khdu

For k=1

Cipher text: HAAHJR HA KHDU
plain text: fyyfhp fy ifbs

For k=2

Cipher text: HAAHJR HA KHDU
plain text: fyyfhp fy ifbs

For k=3

Cipher text: HAAHJR HA KHDU
plain text: exxego ex hear

For k=4

Ciphertext: HAAHJR HA KHDU
plain text: dwwdfn dw gdzq

For k=5

Ciphertext: HAAHJR HA KHDU
plain text: cvvcem cv fcyp

For k=6

Ciphertext: HAAHJR HA KHDU
plain text: buubdl bu ebxo

For k=7

Ciphertext: HAAHJR HA KHDU
plain text: attack at dawn

For k=7, we obtain a meaningful plain text namely **attack at dawn** and hence we are done.

Practical 2

BREAKING THE MONOALPHABETIC SUBSTITUTION CIPHER

STEP 1 : For the given ciphertext in the **PART I** of the experiment page, the first step is to generate ciphertext by clicking on the "Next CipherText" button.

STEP 2 : Calculate frequencies of generated ciphertext by clicking on "Calculate Frequencies in Ciphertext" button

STEP 3 : Copy the generated ciphertext from **PART I** and paste in "Scratchpad" area of **PART II**

STEP 4 : Analyse similarties between "Calculated Frequencies Table" and "English Alphabet Frequencies Table"

STEP 5 : Based on similarities,try to make a frequency based estimation for each character of ciphertext

STEP 6 : Replace characters of CipherText in Scratchpad with a character estimated previously using a **Modify** function of **PART II**

STEP 7 : Based on Hints from Ciphertext in "Scratchpad" area make more replacement of ciphertext characters

STEP 8 : Repeat **Step 7** till you get a meaningful English Text

STEP 9 : Finally, observe the deciphered plaintext in Scratchpad Area,if a meaningful English text is formed cut-and-paste it in the text-field named "Solution Plaintext" of **PART III**. Also enter the final character mapping in the"Solution Key" in **PART III** and click on "Check Answer" button.

STEP 10[OPTIONAL] : Verify that your answer is correct, by encrypting the solution plaintext with your key in **PART IV**.

Example:

CypherText

awbix ildxz kolf a dkzeplld afu zbjjbfm lf bj bz a rwkx iajxpobwwap zdlgbfm a ellgae. jex iajxpobwwap ykxzjblfz awbix afu zex audbjz jl exp ikppxfj buxfjbjt ipbzbz, ildolkfuxu rt exp bfarbwbjt jl pxdxdrxp a olxd. rxnlpx ipahwbfm ahat, jex iajxpobwwap jxwwz awbix jeaj lfx zbux ln jex dkzeplld hbww dagx exp jawwxp afu jex ljexp zbux hbww dagx exp zelpjxp. zex rpxagz lnn jhl obxixz npld jex dkzeplld. lfx zbux dagxz exp zepbfg zdawwxp jeaf xcxp, hebwx afljexp iakzxz exp fxig jl mplh ebme bfjl jex jpxxz, hexpx a obmxlf dbzjagxz exp nlp a zxpoxfj. hbje zldx xnnlpj, awbix rpbfmz expzxwn raig jl exp kzkaw exbmej. zex zjkdrwxz kolf a zdaww xzjajx afu kzxz jex dkzeplld jl pxaie a dlpx aooplopbajx exbmej.

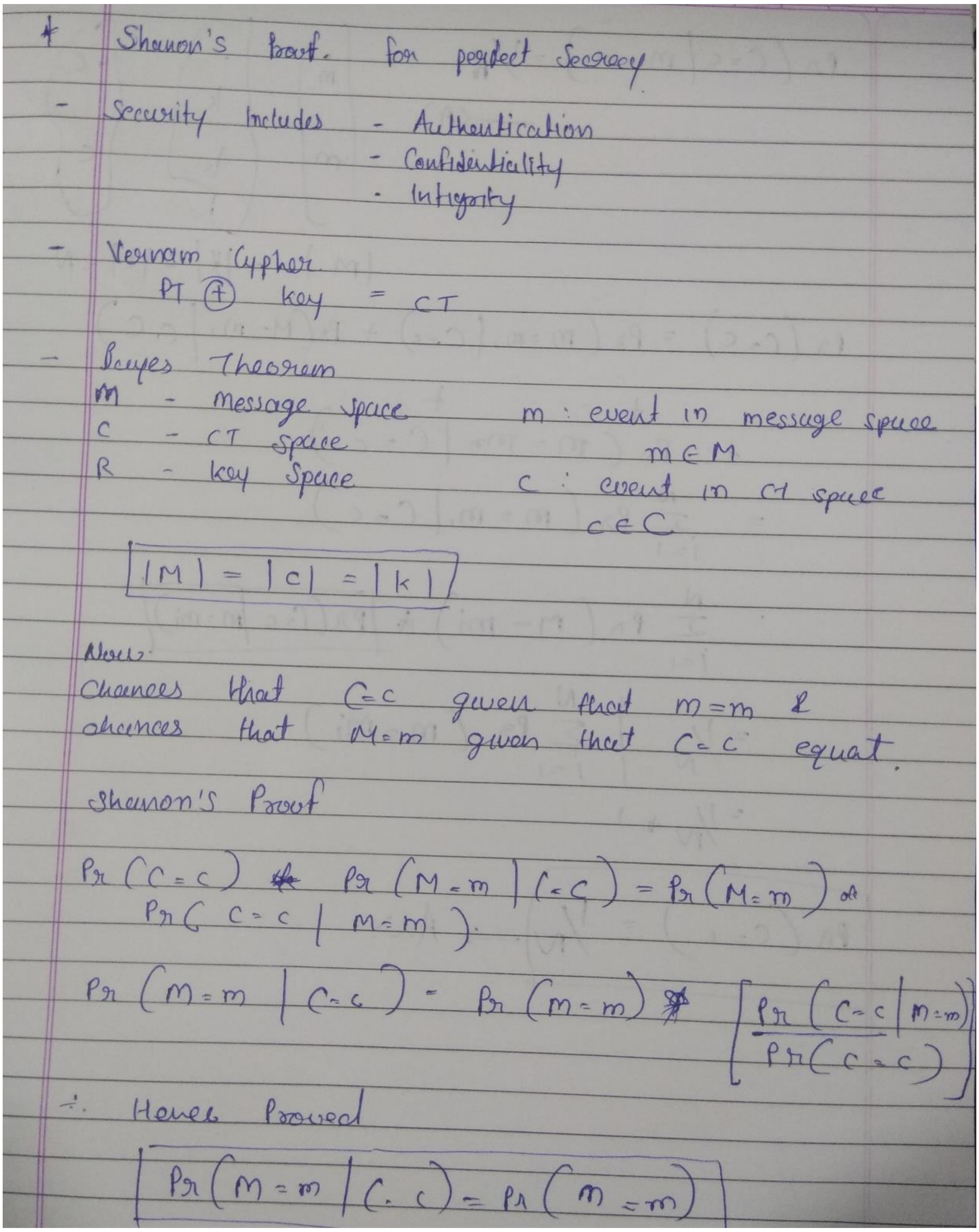
CT	Frequency	PT
a	8.084	A
b	6.854	I
c	0.176	V
d	4.394	M
e	7.206	H
f	4.394	N
g	1.757	K
h	1.582	W
i	2.988	C
j	7.909	T
k	2.636	U
l	7.381	O
m	1.582	G
n	1.582	F
o	2.285	P
p	8.26	R
q	0	J
r	1.582	B
s	0	X
t	0.703	Y
u	1.933	D
v	0	Z
w	5.097	L
x	14.06	E
y	0.176	Q
z	7.381	S

Plain Text :

```
ALICE COMES UPON A MUSHROOM AND SITTING ON IT IS A BLUE CATERPILLAR SMOKING A HOOKAH. THE
CATERPILLAR QUESTIONS ALICE AND SHE ADMITS TO HER CURRENT IDENTITY CRISIS, COMPOUNDED BY
HER INABILITY TO REMEMBER A POEM. BEFORE CRAWLING AWAY, THE CATERPILLAR TELLS ALICE THAT
ONE SIDE OF THE MUSHROOM WILL MAKE HER TALLER AND THE OTHER SIDE WILL MAKE HER SHORTER.
SHE BREAKS OFF TWO PIECES FROM THE MUSHROOM. ONE SIDE MAKES HER SHRINK SMALLER THAN EVER,
WHILE ANOTHER CAUSES HER NECK TO GROW HIGH INTO THE TREES, WHERE A PIGEON MISTAKES HER
FOR A SERPENT. WITH SOME EFFORT, ALICE BRINGS HERSELF BACK TO HER USUAL HEIGHT. SHE
STUMBLES UPON A SMALL ESTATE AND USES THE MUSHROOM TO REACH A MORE APPROPRIATE HEIGHT.
```

Practical 3

SHANON’S PROOF FOR PERFECT SECRECY



* Shanon's Proof. for perfect Secrecy.

- Security Includes
  - Authentication
  - Confidentiality
  - Integrity

- Vernam Cypher

  $PT \oplus key = CT$

- Bayes Theorem

M - message space   m : event in message space $m \in M$

C - CT space   c : event in CT space $c \in C$

K - key Space

$$\boxed{|M| = |C| = |K|}$$

Note:

Chances that C=c given that M=m & chances that M=m given that C=c equal.

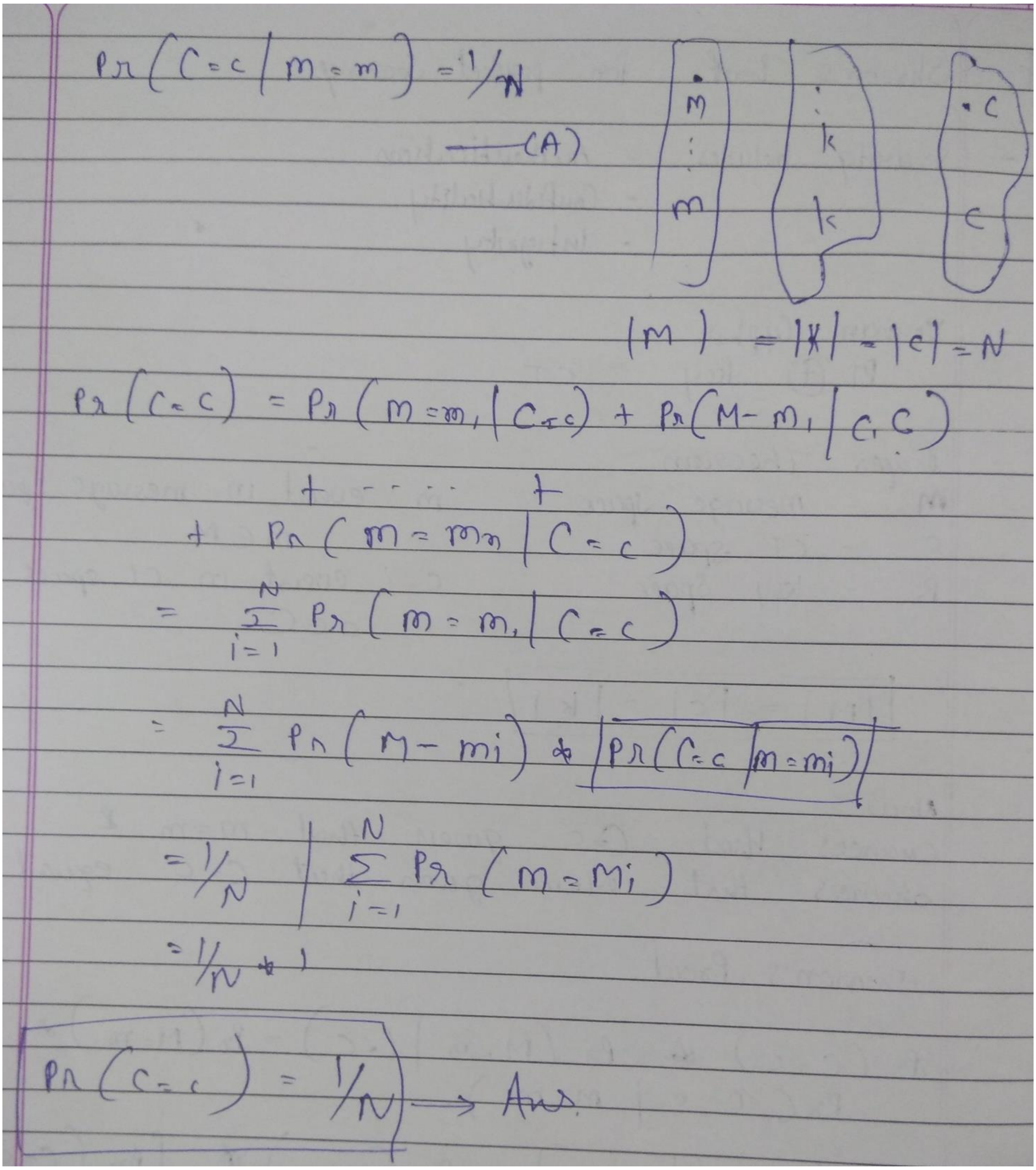
Shanon's Proof

$$Pr(C=c) * Pr(M=m \mid C=c) = Pr(M=m) * Pr(C=c \mid M=m).$$

$$Pr(M=m \mid C=c) = Pr(M=m) * \left[\frac{Pr(C=c \mid M=m)}{Pr(C=c)}\right]$$

∴ Hence Proved

$$\boxed{Pr(M=m \mid C=c) = Pr(M=m)}$$



$$Pr(C=c \mid M=m) = 1/N \quad \text{—— (A)}$$

$$|M| = |K| = |C| = N$$

$$Pr(C=c) = Pr(M=m_1 \mid C=c) + Pr(M-m_1 \mid C, C)$$

$$+ \cdots + $$

$$+ \; Pr(M=m_n \mid C=c)$$

$$= \sum_{i=1}^{N} Pr(M=m_1 \mid C=c)$$

$$= \sum_{i=1}^{N} Pr(M-m_i) * \boxed{Pr(C=c \mid M=m_i)}$$

$$= 1/N \; \Big| \sum_{i=1}^{N} Pr(M=M_i)$$

$$= 1/N * 1$$

$$\boxed{Pr(C=c) = 1/N} \longrightarrow \text{Ans.}$$

Practical 4

SYMMETRIC KEY ENCRYPTION STANDERDS (DES)

Step 1 : Generate Plaintext **m**, **keyA** and **keyB** by clicking on rexpective buttons **PART I** of the simulation page.

Step 2 : Enter generated Plaintext **m** from **PART I** to **PART II** in "Your text to be encrypted/decrypted:" block.

Step 3 : Enter generated **keyA** from **PART I** to **PART II** "Key to be used:" block and click on DES encrpt button to output ciphertext **c1**.This is First Encryption.

Step 4 : Enter generated ciphertext **c1** from **PART II** "Output:" Block to **PART II** in "Your text to be encrypted/decrypted:" block.

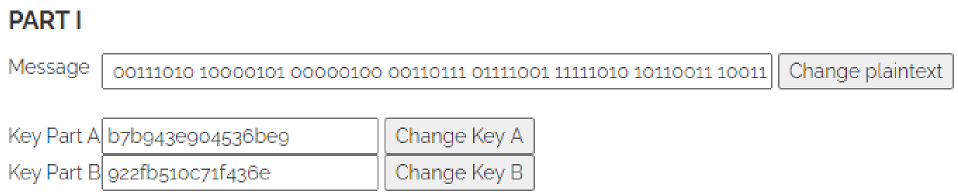
Step 5 : Enter generated **keyB** from **PART I** to **PART II** in "Key to be used:" block and click on DES decrypt button to output ciphertect **c2**.This is Second Encryption.

Step 6 : Enter generated ciphertext **c2** from **PART II** "Output:" block to **PART II** in "Your text to be encrypted/decrypted:" block.

Step 7 : Enter generated **keyA** from **PART I** to **PART II** "Key to be used:" block and click on DES encrpt button to output ciphertext **c3**.This is Third Encryption. As Encryption is done thrice.This Scheme is called triple DES.

Step 7 : Enter generated ciphertext **c3** from **PART II** "Output:" Block to **PART III** "Enter your answer here:" block inorder to verify your Triple DES.

PlainText with Key:



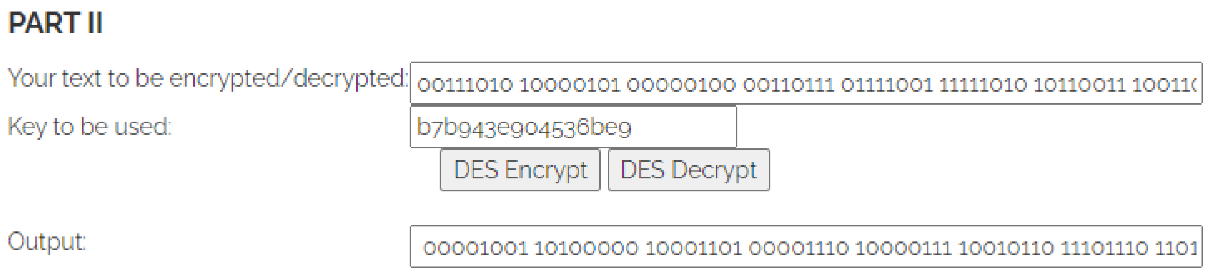
PART I

Message 00111010 10000101 00000100 00110111 01111001 1111010 10110011 10011

Key Part A b7b943e904536be9

Key Part B 922fb510c71f436e

Encryption with key part A



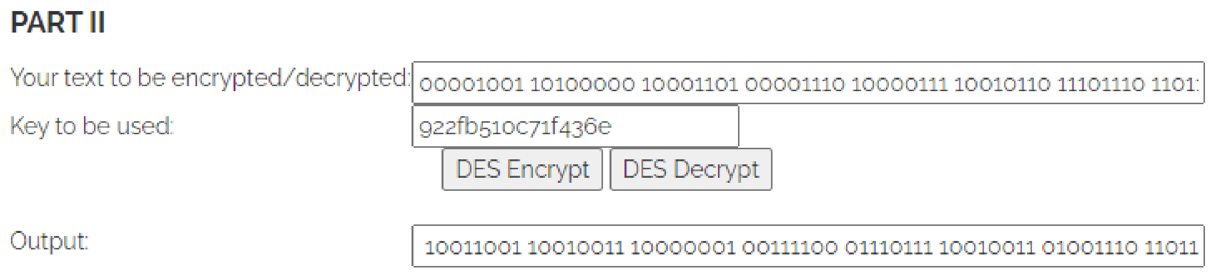
PART II

Your text to be encrypted/decrypted: 00111010 10000101 00000100 00110111 01111001 11111010 10110011 10011

Key to be used: b7b943e904536be9

Output: 00001001 10100000 10001101 00001110 10000111 10010110 11101110 1101

Decryption with key part B



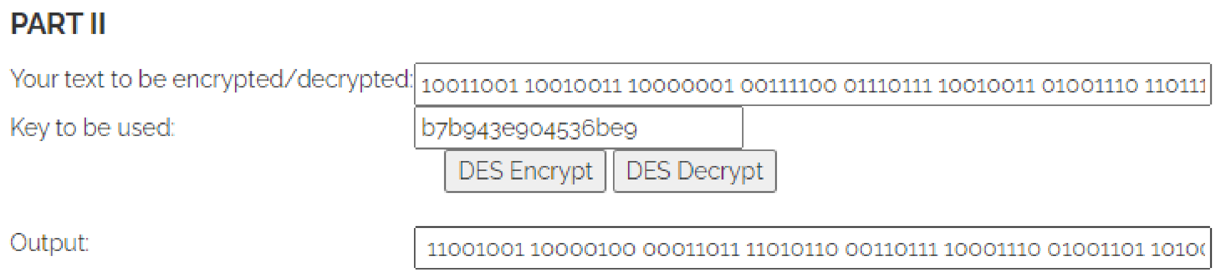
PART II

Your text to be encrypted/decrypted: 00001001 10100000 10001101 00001110 10000111 10010110 11101110 1101

Key to be used: 922fb510c71f436e

Output: 10011001 10010011 10000001 00111100 01110111 10010011 01001110 11011

Encryption with key part A



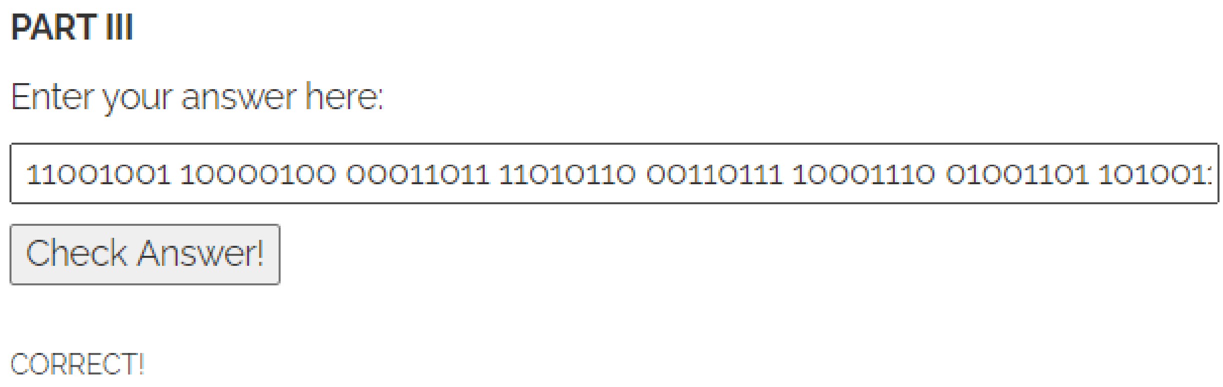
PART II

Your text to be encrypted/decrypted: 10011001 10010011 10000001 00111100 01110111 10010011 01001110 11011

Key to be used: b7b943e904536be9

Output: 11001001 10000100 00011011 11010110 00110111 10001110 01001101 1010

Final Answer



PART III

Enter your answer here:

11001001 10000100 00011011 11010110 00110111 10001110 01001101 101001

CORRECT!

Practical 5

SYMMETRIC KEY ENCRYPTION STANDERDS (AES)

Step I : Choose a mode of operation from *PART I*

Step II : Select KeySize, Plaintext, KeyText, Intialization vector(IV)(for ECB and OFB modes only) and CTR(forctr mode only) in *PART II*

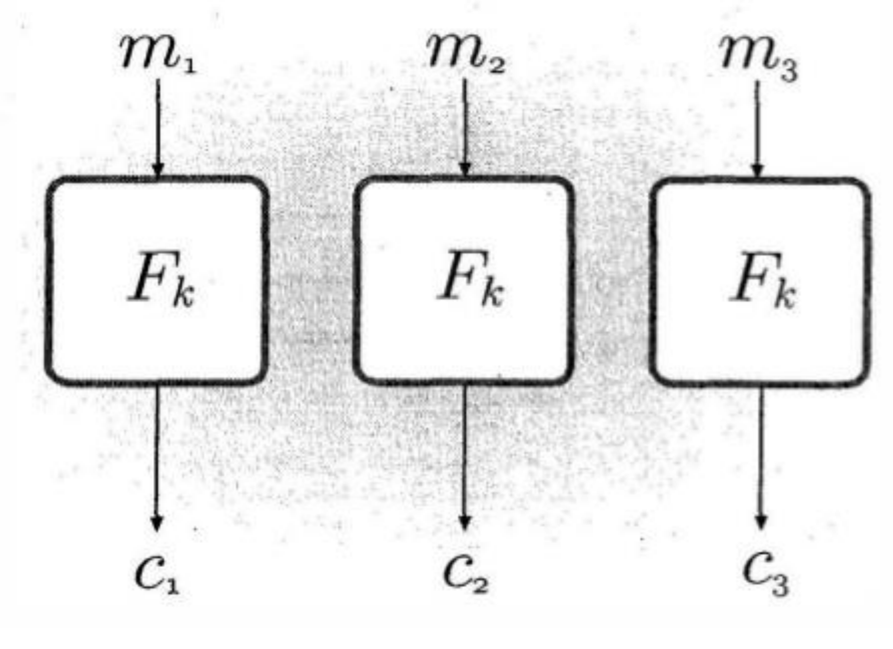
Step III : Whenever necessay use XOR opeartion in *PART III* in accordance with choosen mode of operation

Step IV : Use fuction **F_K** and "Key in hex:" field in *PART IV* should be filled keytext generated in **Step2**

Step V : Fill "Plaintext in hex:" field with approriate value in accordance with choosen mode of opration and click on encrypt button

Step VI : Enter your answer in *PART V* to check your ciphertext

Electronic Code Book(ECB) mode

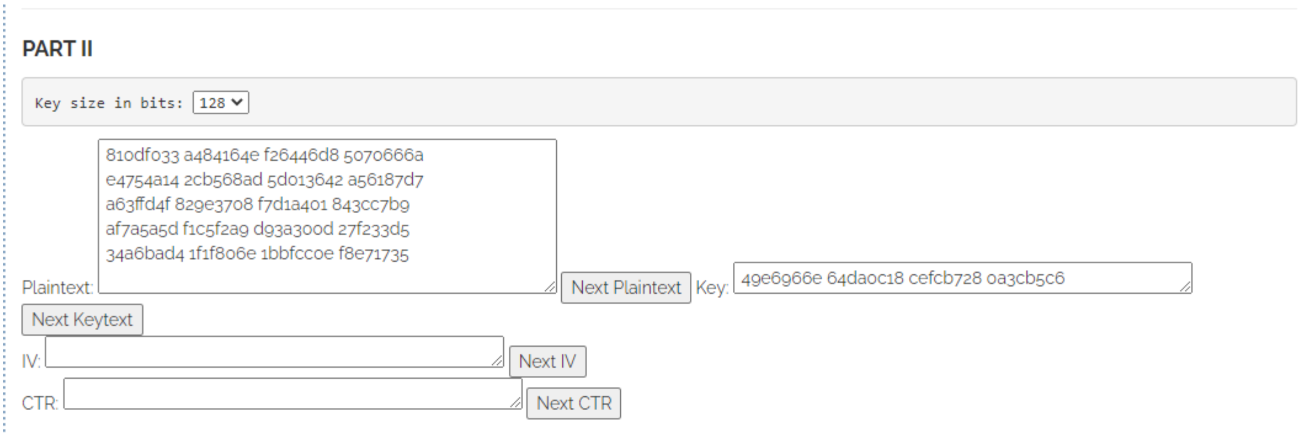


Electronic Code Book(ECB) mode

Example:

PlainText: 810df033 a484164e f26446d8 5070666a
e4754a14 2cb568ad 5d013642 a56187d7
a63ffd4f 829e3708 f7d1a401 843cc7b9
af7a5a5d f1c5f2a9 d93a300d 27f233d5
34a6bad4 1f1f806e 1bbfcc0e f8e71735

Key: 49e6966e 64da0c18 cefcb728 0a3cb5c6



PART II

Key size in bits: 128

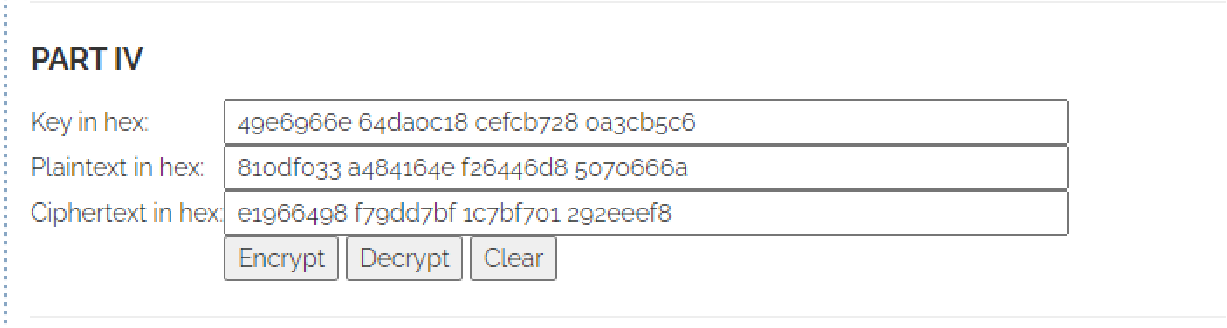
Plaintext:
810df033 a484164e f26446d8 5070666a
e4754a14 2cb568ad 5d013642 a56187d7
a63ffd4f 829e3708 f7d1a401 843cc7b9
af7a5a5d f1c5f2a9 d93a300d 27f233d5
34a6bad4 1f1f806e 1bbfcc0e f8e71735

Key: 49e6966e 64da0c18 cefcb728 0a3cb5c6

IV:

CTR:

Step 1 :



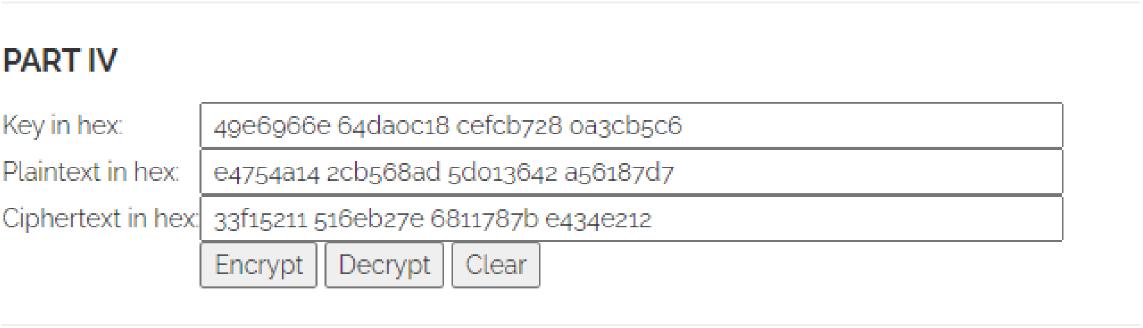
PART IV

Key in hex: 49e6966e 64da0c18 cefcb728 0a3cb5c6

Plaintext in hex: 810df033 a484164e f26446d8 5070666a

Ciphertext in hex: e1966498 f79dd7bf 1c7bf701 292eeef8

Step 2:



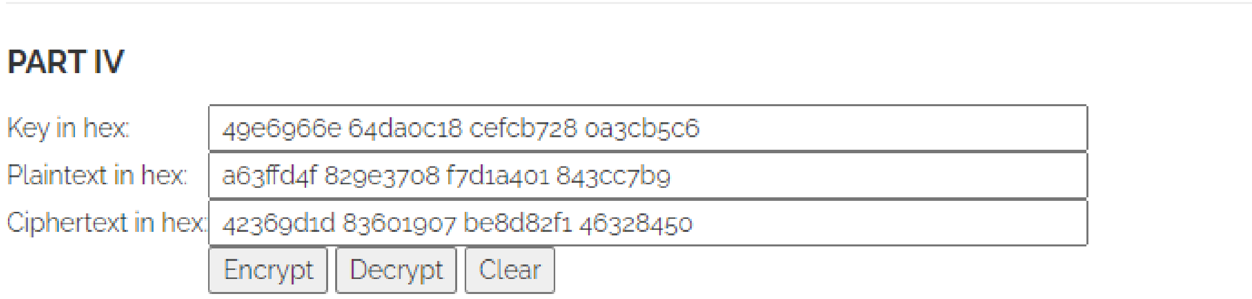
PART IV

Key in hex: 49e6966e 64da0c18 cefcb728 0a3cb5c6

Plaintext in hex: e4754a14 2cb568ad 5d013642 a56187d7

Ciphertext in hex: 33f15211 516eb27e 6811787b e434e212

Step 3:



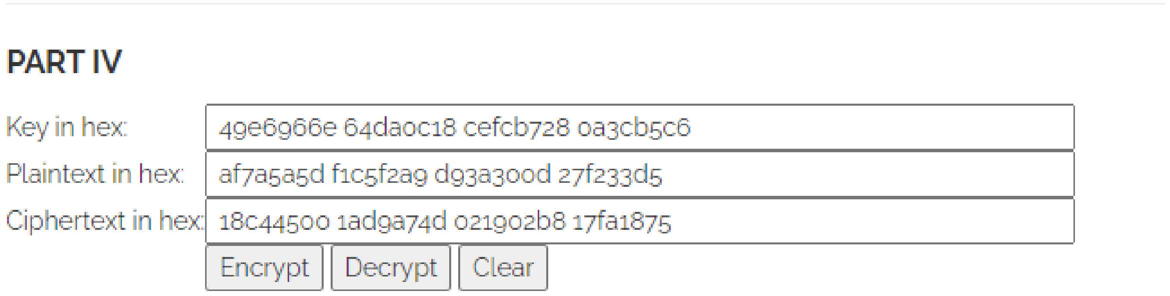
PART IV

Key in hex: 49e6966e 64da0c18 cefcb728 0a3cb5c6

Plaintext in hex: a63ffd4f 829e3708 f7d1a401 843cc7b9

Ciphertext in hex: 42369d1d 83601907 be8d82f1 46328450

Step 4:



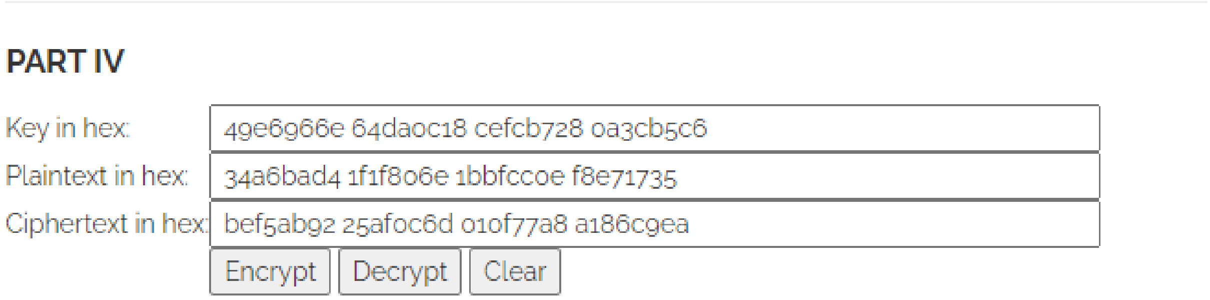
PART IV

Key in hex: 49e6966e 64da0c18 cefcb728 0a3cb5c6

Plaintext in hex: af7a5a5d f1c5f2a9 d93a300d 27f233d5

Ciphertext in hex: 18c44500 1ad9a74d 021902b8 17fa1875

Step 5:



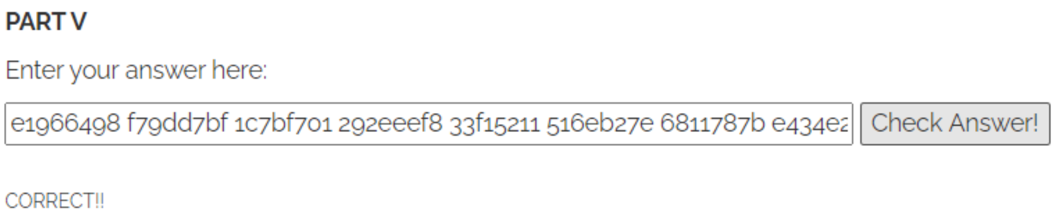
PART IV

Key in hex: 49e6966e 64da0c18 cefcb728 0a3cb5c6

Plaintext in hex: 34a6bad4 1f1f806e 1bbfcc0e f8e71735

Ciphertext in hex: bef5ab92 25af0c6d 010f77a8 a186c9ea

Step 6:



PART V

Enter your answer here:

e1966498 f79dd7bf 1c7bf701 292eeef8 33f15211 516eb27e 6811787b e434e2

CORRECT!!

Cypher Text:

e1966498 f79dd7bf 1c7bf701 292eeef8

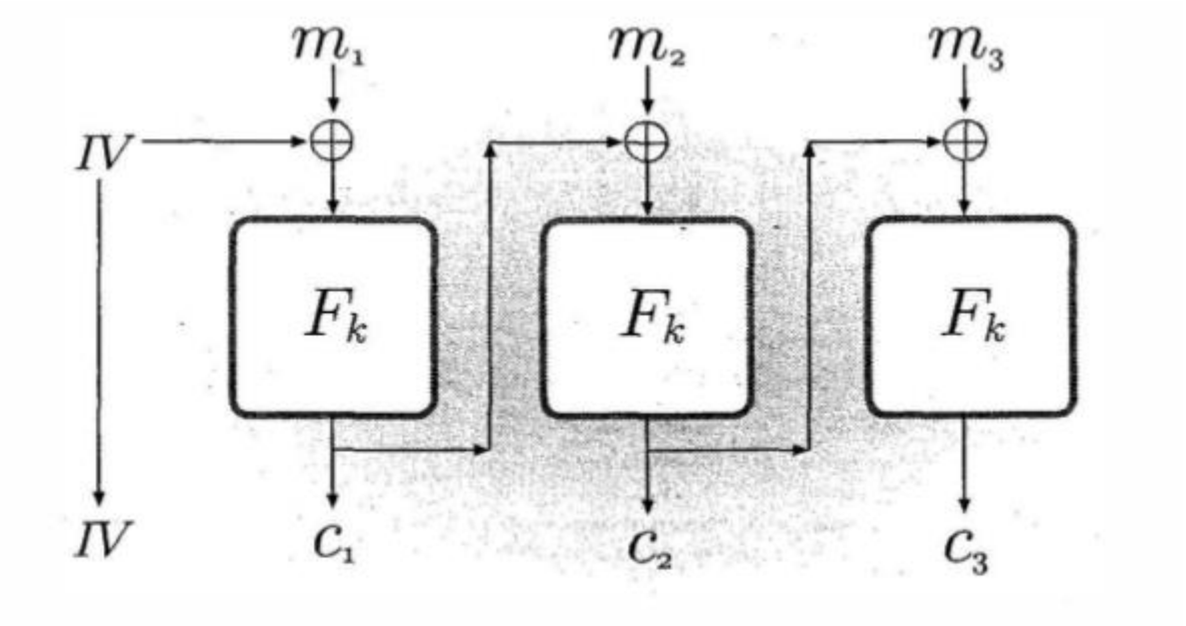
33f15211 516eb27e 6811787b e434e212

42369d1d 83601907 be8d82f1 46328450

18c44500 1ad9a74d 021902b8 17fa1875

bef5ab92 25af0c6d 010f77a8 a186c9ea

Cipher Block Chaining(CBC) mode



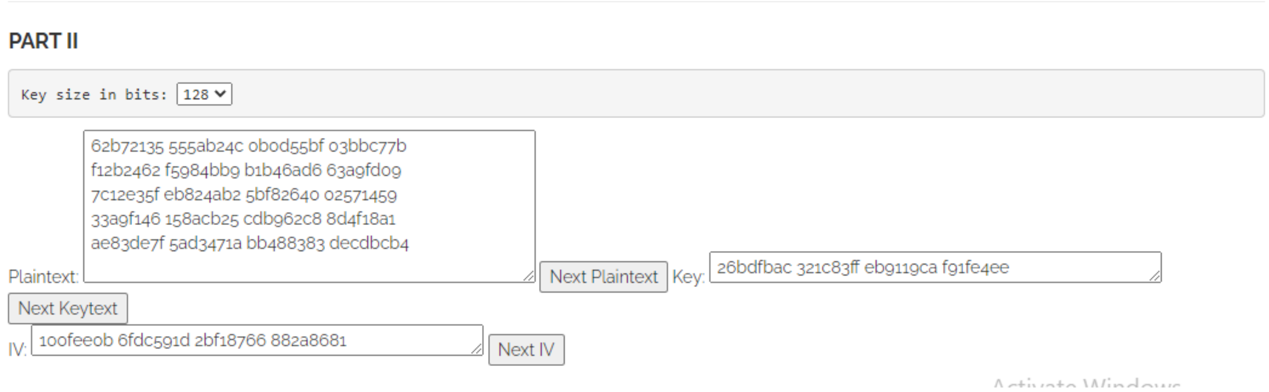
Cipher Block Chaining(CBC) mode

Example:

Plain Text: 62b72135 555ab24c 0b0d55bf 03bbc77b
f12b2462 f5984bb9 b1b46ad6 63a9fd09
7c12e35f eb824ab2 5bf82640 02571459
33a9f146 158acb25 cdb962c8 8d4f18a1
ae83de7f 5ad3471a bb488383 decdbcb4

Key: 26bdfbac 321c83ff eb9119ca f91fe4ee

IV: 100fee0b 6fdc591d 2bf18766 882a8681



PART II

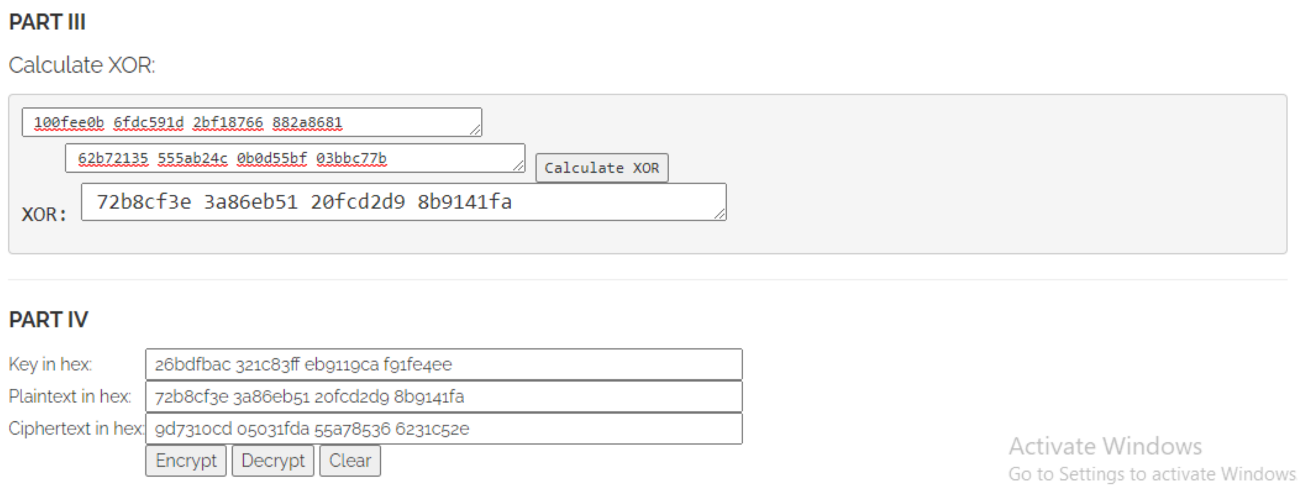
Key size in bits: 128

Plaintext:
62b72135 555ab24c 0b0d55bf 03bbc77b
f12b2462 f5984bb9 b1b46ad6 63a9fd09
7c12e35f eb824ab2 5bf82640 02571459
33a9f146 158acb25 cdb962c8 8d4f18a1
ae83de7f 5ad3471a bb488383 decdbcb4

Key: 26bdfbac 321c83ff eb9119ca f91fe4ee

IV: 100fee0b 6fdc591d 2bf18766 882a8681

Step 1:



PART III

Calculate XOR:

100fee0b 6fdc591d 2bf18766 882a8681

62b72135 555ab24c 0b0d55bf 03bbc77b

XOR : 72b8cf3e 3a86eb51 20fcd2d9 8b9141fa

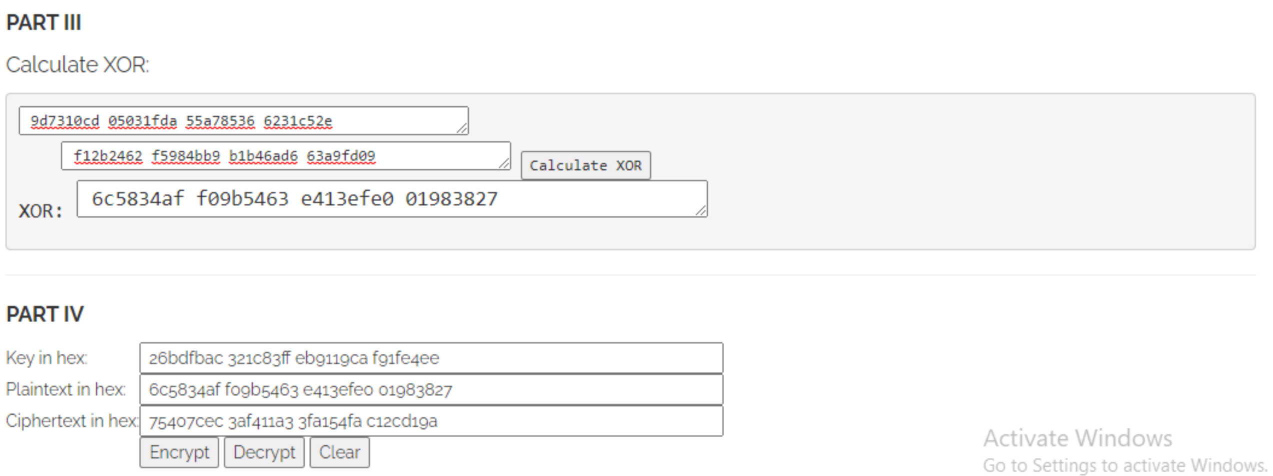
PART IV

Key in hex: 26bdfbac 321c83ff eb9119ca f91fe4ee

Plaintext in hex: 72b8cf3e 3a86eb51 20fcd2d9 8b9141fa

Ciphertext in hex: 9d7310cd 05031fda 55a78536 6231c52e

Step 2:



PART III

Calculate XOR:

9d7310cd 05031fda 55a78536 6231c52e

f12b2462 f5984bb9 b1b46ad6 63a9fd09

XOR : 6c5834af f09b5463 e413efe0 01983827

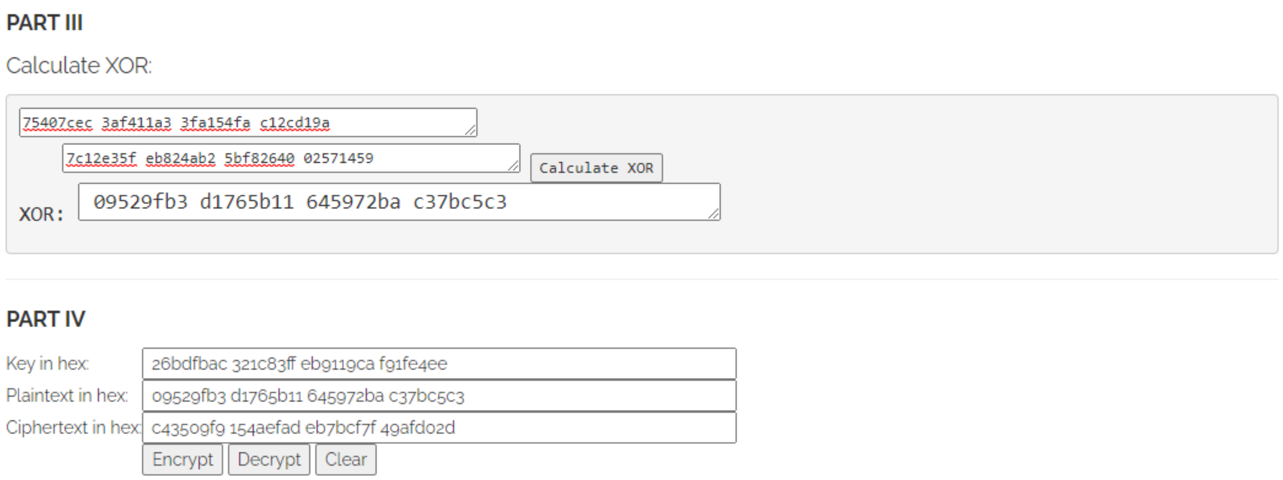
PART IV

Key in hex: 26bdfbac 321c83ff eb9119ca f91fe4ee

Plaintext in hex: 6c5834af f09b5463 e413efe0 01983827

Ciphertext in hex: 75407cec 3af411a3 3fa154fa c12cd19a

Step 3:



PART III

Calculate XOR:

75407cec 3af411a3 3fa154fa c12cd19a

7c12e35f eb824ab2 5bf82640 02571459

XOR : 09529fb3 d1765b11 645972ba c37bc5c3

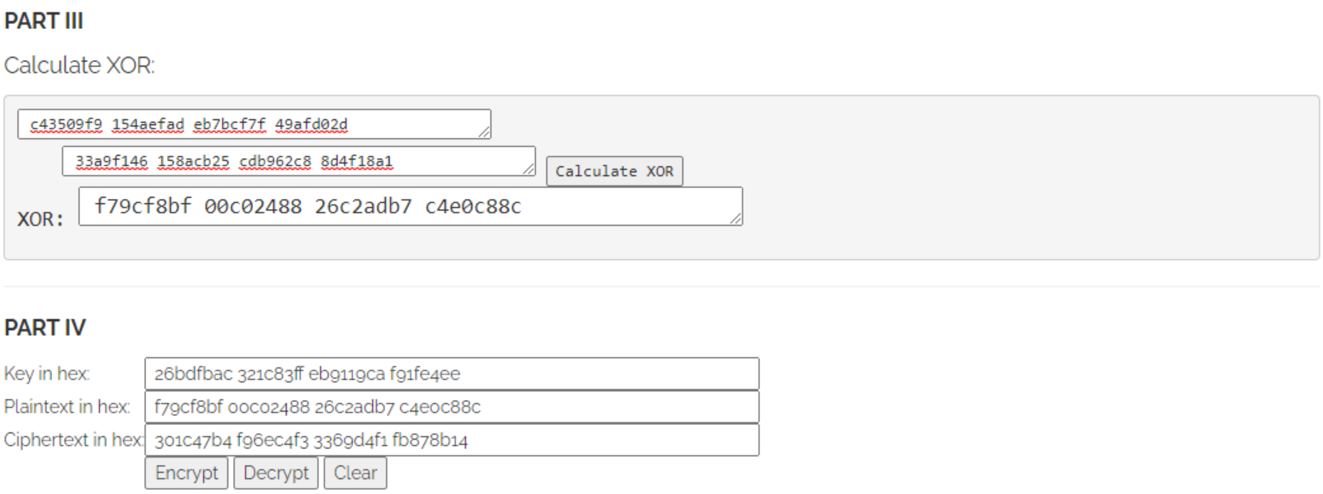
PART IV

Key in hex: 26bdfbac 321c83ff eb9119ca f91fe4ee

Plaintext in hex: 09529fb3 d1765b11 645972ba c37bc5c3

Ciphertext in hex: c43509f9 154aefad eb7bcf7f 49afd02d

Step 4:



PART III

Calculate XOR:

c43509f9 154aefad eb7bcf7f 49afd02d

33a9f146 158acb25 cdb962c8 8d4f18a1

XOR: f79cf8bf 00c02488 26c2adb7 c4e0c88c

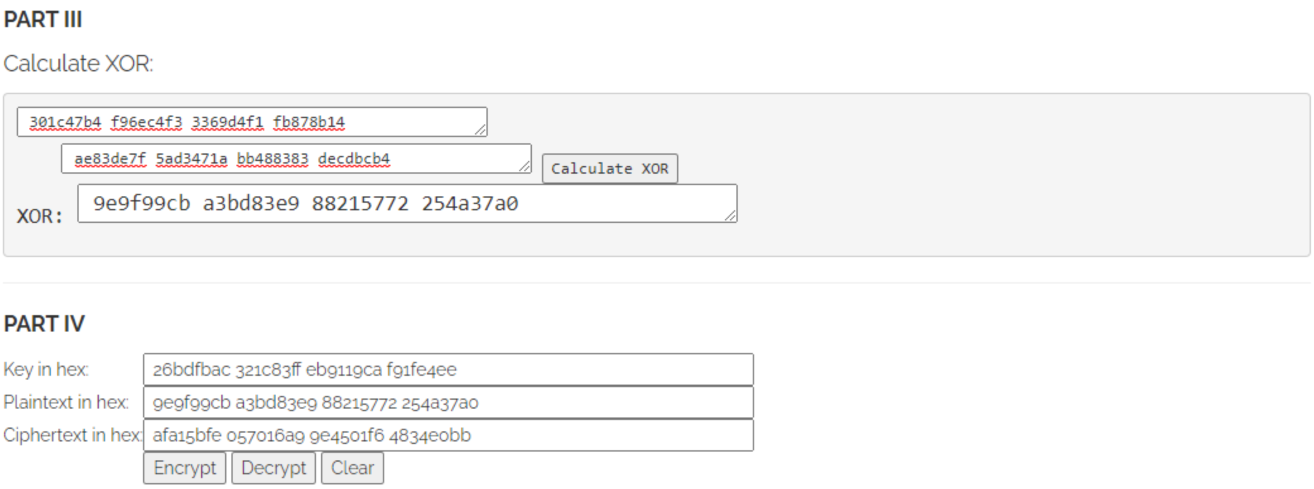
PART IV

Key in hex: 26bdfbac 321c83ff eb9119ca f91fe4ee

Plaintext in hex: f79cf8bf 00c02488 26c2adb7 c4e0c88c

Ciphertext in hex: 301c47b4 f96ec4f3 3369d4f1 fb878b14

Step 5:



PART III

Calculate XOR:

301c47b4 f96ec4f3 3369d4f1 fb878b14

ae83de7f 5ad3471a bb488383 decdbcb4

XOR: 9e9f99cb a3bd83e9 88215772 254a37a0

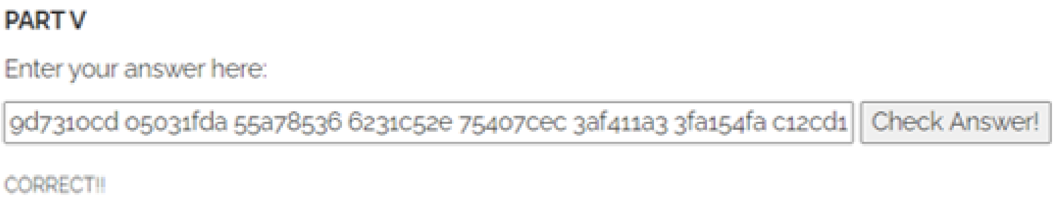
PART IV

Key in hex: 26bdfbac 321c83ff eb9119ca f91fe4ee

Plaintext in hex: 9e9f99cb a3bd83e9 88215772 254a37a0

Ciphertext in hex: afa15bfe 057016a9 9e4501f6 4834e0bb

Step 6:



PART V

Enter your answer here:

9d7310cd 05031fda 55a78536 6231c52e 75407cec 3af411a3 3fa154fa c12cd1

CORRECT!!

Cypher Text:

9d7310cd 05031fda 55a78536 6231c52e

75407cec 3af411a3 3fa154fa c12cd19a

c43509f9 154aefad eb7bcf7f 49afd02d

301c47b4 f96ec4f3 3369d4f1 fb878b14

afa15bfe 057016a9 9e4501f6 4834e0bb

Practical 6

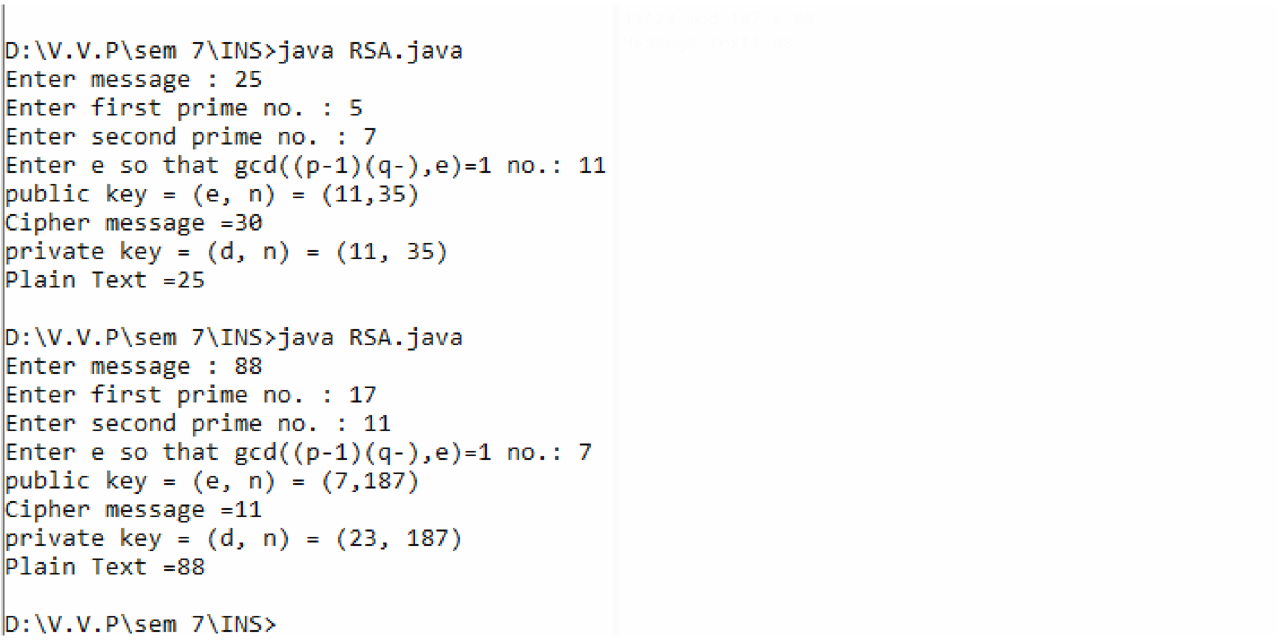
WRITE A PROGRAM TO IMPLEMENT RSA.

```
import java.util.Scanner;
import java.lang.Math;
class RSA{
public static void main(String[] args) {
Scanner s = new Scanner(System.in);
System.out.print("Enter message : ");
int message = s.nextInt();
System.out.print("Enter first prime no. : ");
int p = s.nextInt();
System.out.print("Enter second prime no. : ");
int q = s.nextInt();
int n = p*q;
int o = (p-1)*(q-1);
System.out.print("Enter e so that gcd((p-1)(q-),e)=1 no.: ");
int e = s.nextInt();

int cipherMsg = encrypt(message,n,e);
System.out.println("public key = (e, n) = ("+ e+","+n +")" );
System.out.println("Cipher message ="+cipherMsg);

int decPlainMsg = decrypt(cipherMsg,n,o,e);
System.out.println("Plain Text ="+decPlainMsg);
}
public static int encrypt(int msg,int n,int e){
int cm = 1;
for(int i = 0 ; i < e ; i++){
cm = (cm *msg) %n;
}
return cm;
}
public static int decrypt(int c,int n,int o,int e){
int d = findGCD(o,e);
System.out.println("private key = (d, n) = ("+ d +", "+n +")" );
int pm = 1;
//System.out.println(c +" "+d + " "+Math.pow(c,d) + " " +n );
for(int i = 0 ; i < d ; i++){
pm = (pm *c) %n;
}
return pm;
}
public static int findGCD(int o,int e){
int q ,temp,t0=0,t1=1,t=0;
while(e != 1){
q =o/e;
temp = o;
o = e;
e = temp % e;
t = t0 - q*t1;
t0 = t1;
t1 = t;
}
return t;
}
}
```

OUTPUT:



```
D:\V.V.P\sem 7\INS>java RSA.java
Enter message : 25
Enter first prime no. : 5
Enter second prime no. : 7
Enter e so that gcd((p-1)(q-),e)=1 no.: 11
public key = (e, n) = (11,35)
Cipher message =30
private key = (d, n) = (11, 35)
Plain Text =25

D:\V.V.P\sem 7\INS>java RSA.java
Enter message : 88
Enter first prime no. : 17
Enter second prime no. : 11
Enter e so that gcd((p-1)(q-),e)=1 no.: 7
public key = (e, n) = (7,187)
Cipher message =11
private key = (d, n) = (23, 187)
Plain Text =88

D:\V.V.P\sem 7\INS>
```