Name - Umang Patel
Roll No. - 2193271
Subject - Information Security

## Assignment - 1.

Chinese Remainder Theorem

**Problem Statement:** Implement a number theory such as Chinese remainder Theorem.

**Objective:** To Study & implement Chinese Reminder Theorem.

**Theory:** Chinese Reminder Theorem is used to solve set of congruent equations with one variable but different modulus, which are relatively prime.

$$x = a1 \bmod m1.$$
$$x = a2 \bmod m2.$$
$$x = a3 \bmod m3.$$
$$\ldots\ldots$$
$$x = ak \bmod mk.$$

The Chinese Reminder Theorem States that the above equations have unique solution if the moduli are relatively prime. Below are the steps needed to follow to solve set of congruent equations using Chinese Reminder Theorem

Scanned with CamScanner

Step 1: Find $M = m1 \times m2 \times m3 \ldots mk$ where M is common modulus.

Step 2: Find $M1 = m/m1$, $m2 = M/m2$ & so on.

Step 3: Find multiplicative inverse for M1, M2 & so on

Step 4: Put the values in the below equation to solve for X.

$$X = (a1 \times m1 \times M1^{-1} + a2 \times M2 \times M2^{-1} + a3 \times M3 \times M3^{-1}) \bmod M.$$

Example:

$X = 4 \bmod 5$

$X = 6 \bmod 8$

$X = 9 \bmod 9$

Step 1: $M = 5 * 8 * 9 = 360$

Step 2: $M1 = M/m1 = 360/5 = 72$

$M2 = m/m2 = 360/8 = 45$

$M3 = M/m3 = 360/9 = 840$

Step 3:- Find M1 inverse, Solver for GCD (M1, m1) using Extended Euclidean Algorithm

GCD (5, 72)

| q | r1 | r2 | r | t1 | t2 | t |
|---|---|---|---|---|---|---|
| 0 | 5 | 72 | 5 | 0 | 1 | 0 |
| 14 | 72 | 5 | 2 | 1 | 0 | 1 |
| 2 | 5 | 2 | 1 | 0 | 1 | -2 |

The inverse value cannot be negative, so add modulus into it to make it positive.

M1 inverse = $-2 + 5 = 3$.

Scanned with CamScanner

Find M2 inverse, Solve for GCD (m2, M2) using Extended Euclidean Algorithm,

GCD(8,45)

| q | r1 | r2 | r | t1 | t2 | t |
|---|---|---|---|---|---|---|
| 0 | 8 | 45 | 8 | 0 | 1 | 0 |
| 5 | 45 | 8 | 5 | 1 | 0 | 1 |
| 1 | 8 | 5 | 3 | 0 | 1 | -1 |
| 1 | 5 | 3 | 2 | 1 | -1 | 2 |
| 1 | 3 | 2 | 1 | -1 | 2 | -3 |

To find M3 inverse, for GCD(M3, M3)

GCD (9,40)

| q | r1 | r2 | r | t1 | t2 | t |
|---|---|---|---|---|---|---|
| 0 | 9 | 40 | 9 | 0 | 1 | 0 |
| 4 | 40 | 9 | 4 | 1 | 0 | 1 |
| 2 | 9 | 4 | 1 | 0 | 1 | -2 |

Step 4: Put values in the below equation solve for x.

$$X = (a1 \times M1 \times M1^{-1} + a2 \times M2 \times M2^{-1} + a3 \times M3 \times M3^{-1}) \mod M.$$

$X = (4*72*3 + 6*45*5 + 8*40*7) \mod 360.$

$X = (864 + 1350 + 2240) \mod 360$

$x = 4454 \mod 360$

$x = 134.$

Scanned with CamScanner

Applications:

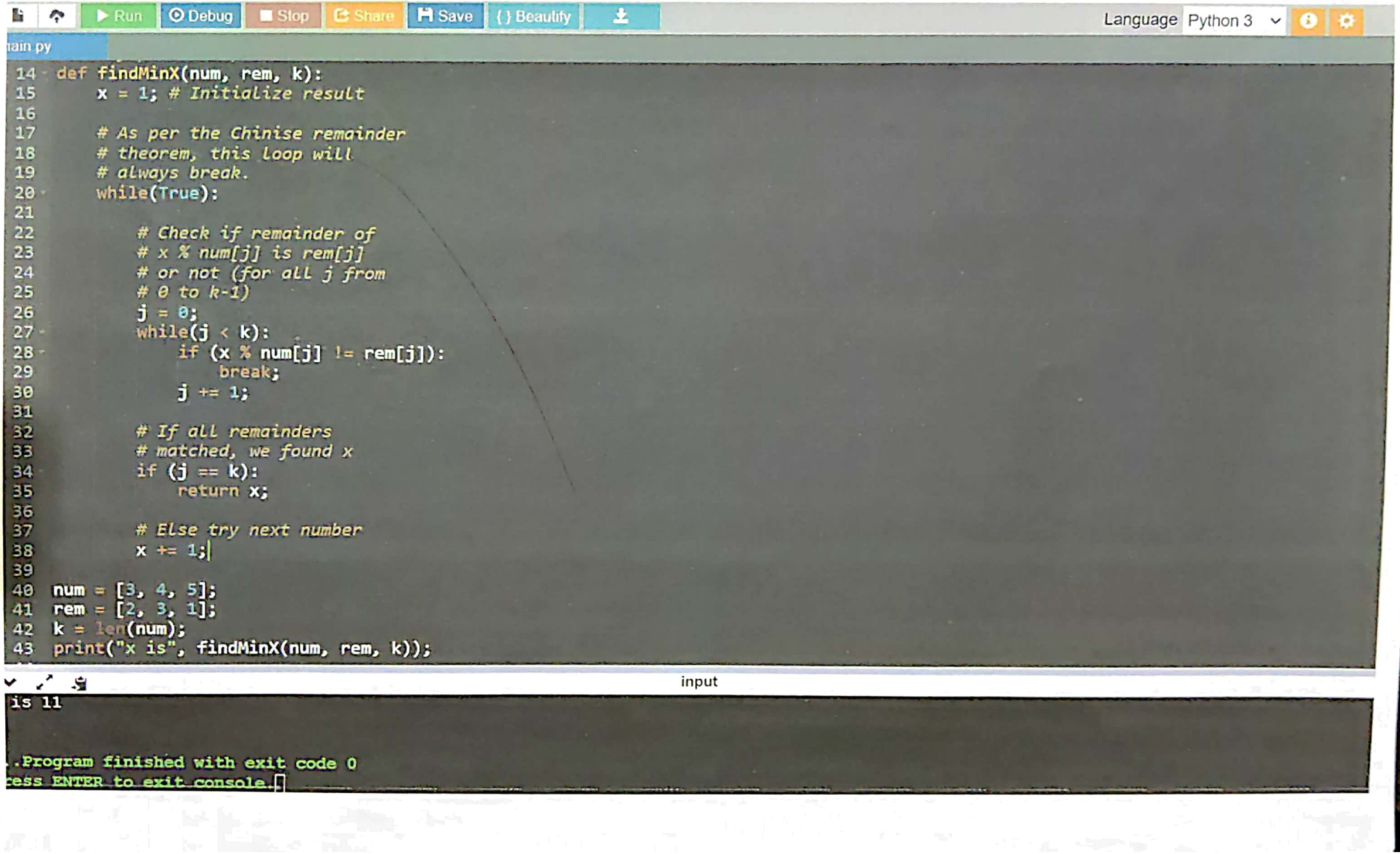
The Chinese Reminder Theorem has several applications in cryptography one is to solve the quadratic congruence & the other is to represent very large number in terms of list of small integers.

Conclusion:

We have studied & implemented Chinese Reminder Theorem.

Whaya
13/10/22

Scanned with CamScanner




Scanned with CamScanner

Assignment - 2.

Extended Euclidian Algorithm.

Problem Statement : Implement Euclidiean & Extended Euclidean algorithm to find out GCD & solve the inverse mod problem.

Objectives: To Study Euclidian & Extended Euclidian algorithm.

Theory: The extended Euclidean algorithm is an extension to the Euclidean algorithm. Besides finding the greatest common divisor of integers a & b, as the Euclidean algorithm does, it also finds integers x & y

$ax + by = \gcd(a,b)$ or $sa + tb = \gcd(a,b)$.

The extended Euclidean algorithm is particularly useful when a & b are coprime, since x is the multiplicative inverse of a modulo b, & y is the multiplicative inverse of b modulo a.

Scanned with CamScanner

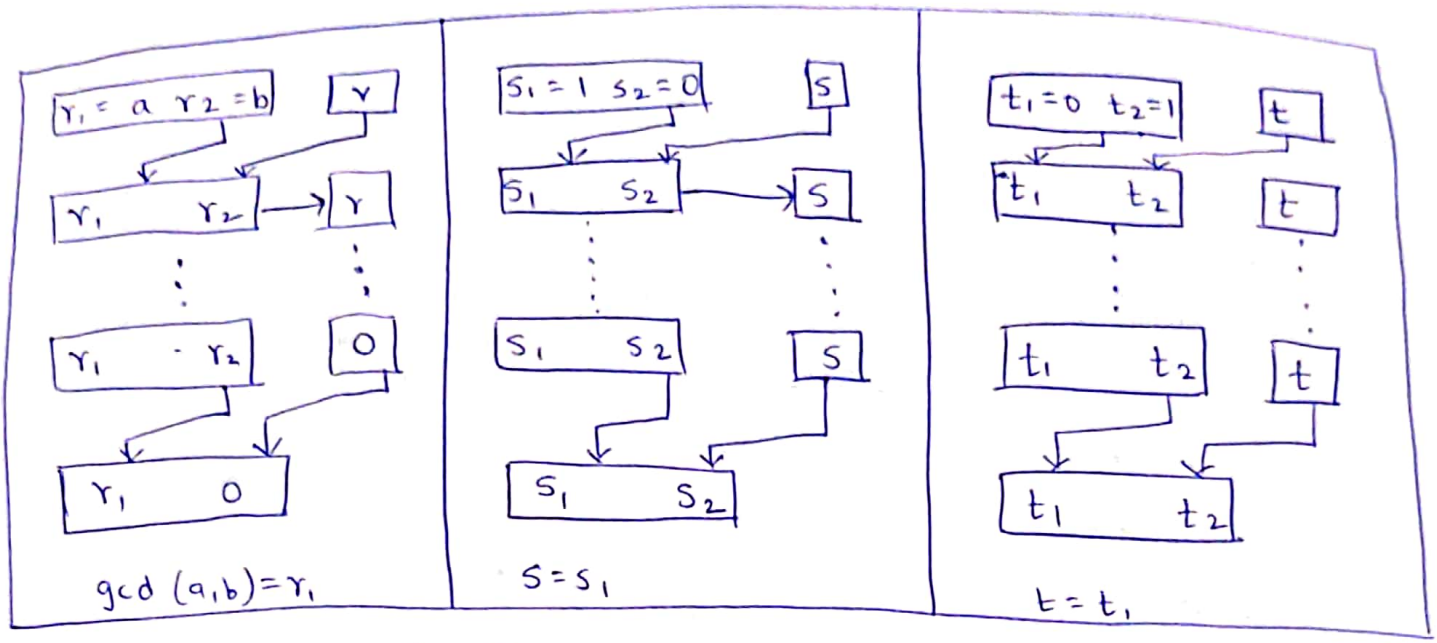



Fig:- Extended Euclid's Algorithm Process

Algorithm:- Extended the algorithm to compute the integer coefficients x & y such that gcd (a,b) = ax + by

Extended - Euclid (a,b).

$r_1 \leftarrow a$ ; $r_2 \leftarrow b$

$s_1 \leftarrow 1$; $s_2 \leftarrow 0$;

$t_1 \leftarrow 0$; $t_2 \leftarrow 1$;

while ($r_2 > 0$)

{

$q \leftarrow r_1 / r_2$ ;

$r \leftarrow r_1 - q \times r_2$ ;

$r_1 \leftarrow r_2$ ; $r_2 \leftarrow r$ ; (updating r's)

$s \leftarrow s_1 - q \times s_2$

$s_1 \leftarrow s_2$ ; $s_2 \leftarrow s$; (updating s's)

$t \leftarrow t_1 - q \times t_2$ ; (updating t's)

$t_1 \leftarrow t_2$ ; $t_2 \leftarrow t$;

}

gcd (a,b) $\leftarrow r_1$ ; $s \leftarrow s_1$ ; $t \leftarrow t_1$

Scanned with CamScanner

Example:

GCD( 161, 28)

| q | r1 | r2 | r | s1 | s2 | s | t1 | t2 | t |
|---|---|---|---|---|---|---|---|---|---|
| 5 | 161 | 28 | 21 | 1 | 0 | 1 | 0 | 1 | -5 |
| 1 | 28 | 21 | [7] | 0 | 1 | [-1] | 1 | -5 | [6] |
| 3 | 21 | 7 | 0 | 1 | -1 | 4 | -5 | 6 | -23 |
| | 7 | 0 | | -1 | 4 | | 6 | -23 | |

Here GCD value we are geeting as 7. Value of s is -1 & value of t is 6. If we put these values into equation,

$$ax + by = gcd(a,b)$$

$$161 * (-1) + 28 * (6) = 7.$$

This satisfies the equation for Extended Euclidean Algorithm.

Conclusion:-

We have studied & implemented the Extended Euclidean algorithm.

Mahajan
13/10/22

Scanned with CamScanner

Run | Debug | Stop | Share | Save | { } Beautify | Language: Python 3

main.py

```python
# Python program to demonstrate working of extended
# Euclidean Algorithm

# function for extended Euclidean Algorithm
# Name- Umang Patel


def gcdExtended(a, b):

    # Base Case
    if a == 0:
        return b, 0, 1

    gcd, x1, y1 = gcdExtended(b % a, a)

    # Update x and y using results of recursive
    # call
    x = y1 - (b//a) * x1
    y = x1

    return gcd, x, y


# Driver code
a, b = 35, 15
g, x, y = gcdExtended(a, b)
print("gcd(", a, ",", b, ") = ", g)

```

input

```
cd( 35 , 15 ) =  5


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

Scanned with CamScanner

Assignment - 3

RSA Algorithm.

Problem Statement: Implement RSA public key cryptosystem for key generation & cipher verification.

Objectives: To understand,

1) Public key algorithm.
2) RSA algorithm.
3) Concept of Public key & Private key.

Theory:-

- .Public key Algorithm:

Asymmetric algorithms relay on one key for encryption & a a different but related key for decryption.

These algorithms have the following characteristics.

- It is computationally infeasible to determine the decryption key given only knowledge of the cryptographic algorithm and the encryption key.

In addition, some algorithms, such as RSA, also exhibit the following characteristics:

- Either of the two related keys can be used for encryption, with other used for decryption.

Scanned with CamScanner

A public key encryption scheme has six ingredients :-

- Plaintext : This is readable message or data that is fed into the algorithm as input.
- Encryption algorithm :- The encryption algorithm performs various transformations on the plaintext.
- Public & Private Key : This is a pair of keys that have been selected so that if one is used for encryption, the other is used for decryption.
- Cipher Text : - This is the scrambled message produced as output. It depends on the plaintext & the key. For a given message, two different keys will produce two different cipher texts.
- Decryption algorithm: This algorithm accepts the ciphertext & the matching key & produces the original plaintext.

# The RSA Algorithm :-

The scheme developed by Rivest, Shamir & Aldeman makes use of an expression with exponentials. Plaintext is encrypted in blocks, with each block having a binary values less than some number n. That is the block size must be less than or equal to log 2 (n) ; in practice the block size is 1 bits, Encryption & decryption are of the following form,

$$C = M^e \bmod n$$

$$M = C^d \bmod n.$$

Scanned with CamScanner

Both Sender & receiver must know the value of n. The Sender known the value of e, & only the receiver, knows the value of d.

Thus, this is a public key encryption algorithm. with a public key of PU = {e, n} & a private key of PR = {d, n}, for this to be satisfactory, the following requirements must meet:

1) It is possible to find values of e, d, n.
2) It is relatively easy to calculate $M^e \bmod n$ & $C^d \bmod n$ for all values of $M < n$.
3) It is feasible to determine d given e & n.

| Key Generation. | |
|---|---|
| Select p, q | p & q both prime, $p \neq q$ |
| Calculate $n = p * q$ | |
| Calculate $\phi(n) = (p-1)(q-1)$ | |
| Select integer e | $\gcd(\phi(n), e) = 1;\ 1 < e < \phi(n)$ |
| Calculate d | $d = e^{(-1)} \bmod \phi(n)$ |
| Public key | $PU = \{e, n\}$ |
| Private key | $PR = \{d, n\}$ |

| Encryption | |
|---|---|
| Plaintext | $M < n$ |
| Ciphertext | $C = M^e \bmod n$ |

| Decryption | |
|---|---|
| Ciphertext | C |
| Plaintext | $M = C^d \bmod n$ |

fig:- The RSA Algorithm.

Scanned with CamScanner

Example :

P = 7, Q = 13, M = 10.

Step 1 : $n = 3 * 11 = 33$

Step 2 : $\phi(n) = 2 * 10 = 20$

Step 3 : Select e, such that gcd $(\phi(n), e) = 1$, gcd $(20, 3) = 1$, so we can select $e = 3$.

Step 4 : To calculate d, solve for gcd $(\phi(n), e)$ using extended Euclid's algorithm & pick up the value of t.

| q | r1 | r2 | r | t1 | t2 | t |
|---|---|---|---|---|---|---|
| 6 | 20 | 3 | 2 | 0 | 1 | -6 |
| 1 | 3 | 2 | 1 | 1 | -6 | 7 |

$d = 7$.

Step 5 : for Encryption,

$C = M^e \bmod n$

$= 2^3 \bmod 33$

$= 8 \bmod 33$

$= 8$.

Step 6 : for Decryption,

$M = C^d \bmod n$

$= 8^7 \bmod 33$

$= ((8^2 \bmod 33)(8^2 \bmod 33)(8^2 \bmod 33)(8^1 \bmod 33)) \bmod 33$

$= (31 * 31 * 8 * 31) \bmod 33$

$= (961 \bmod 33)(248 \bmod 33) \bmod 33$.

$= 68 \bmod 33$

$= 2$.

Scanned with CamScanner

Advantage:

1) Easy to implement.

Disadvantage:

1) Anyone can annouze the public key.

Algorithm:

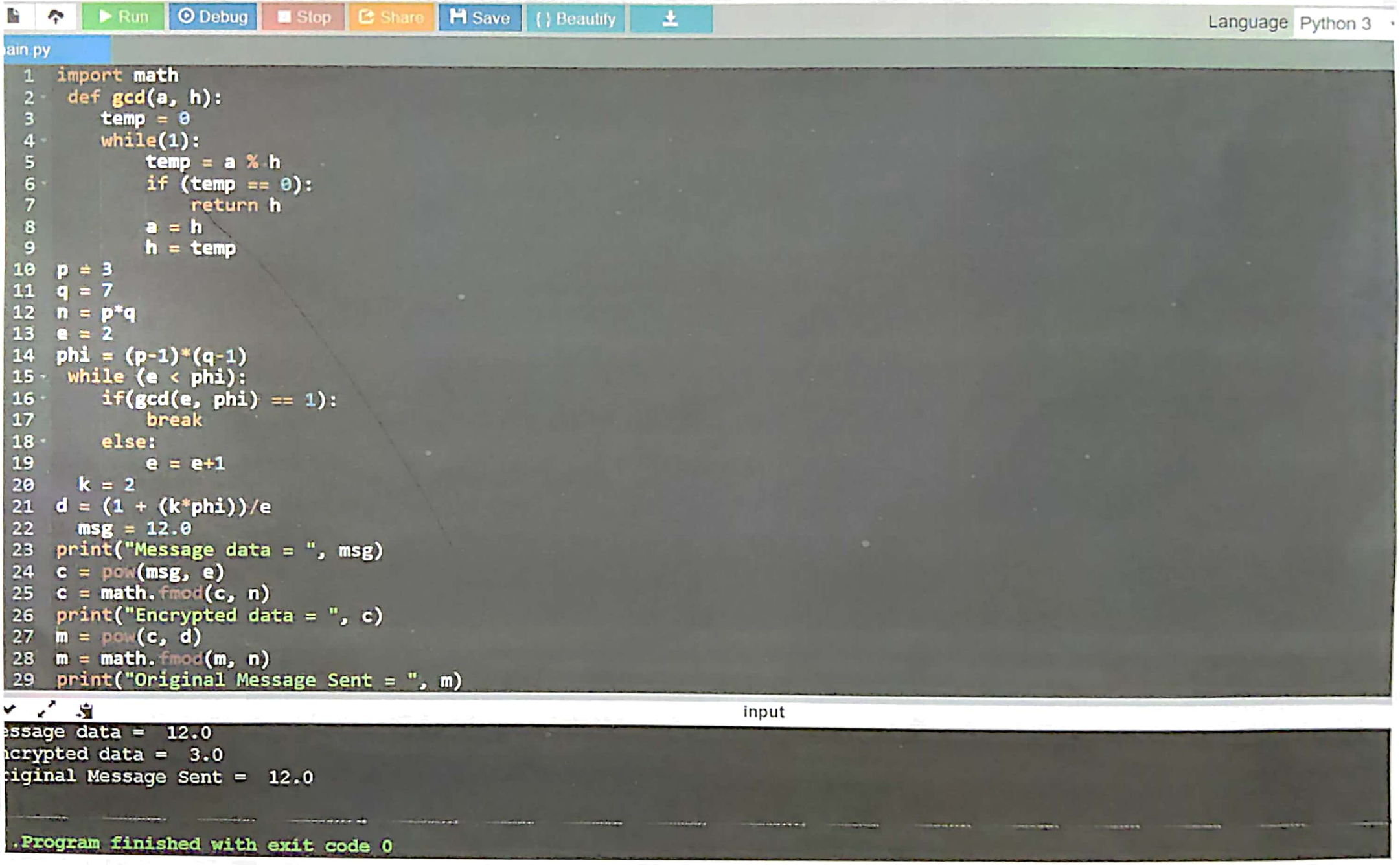
1) Start
2) Input two prime numbers p & q.
3) Calculate $n = pq$.
4) Calculate $\phi(n) = (p-1)(q-1)$.
5) Input value of e.
6) Determine d.
7) Determine PU & PR.
8) Take input plaintext.
9) Encrypt the plaintext & Show output.
10) Stop.

Conclusion:

We have studied & implemented the public key algorithm that is RSA algorithm.

13/10/22

Scanned with CamScanner




Scanned with CamScanner

Name - Vnang Patel
Roll No. - 2193271
Subject - I.S
Class - LY IS-3.

Assignment - 04.

Problem Statement :- Implement Diffie Hellman Key exchange algorithm for secret key generation & distribution of public key.

Objective :-

1) To learn the basics of key management.
2) To study & implement Diffie Hellman Key exchange algorithm.

Theory :-

The purpose of the algorithm is to enable two users to securely exchange a key that can then be used for subsequent encryption of messages. The algorithm itself is limited to the exchange & secret of secret values.

Scanned with CamScanner

Algorithm :-

- There are two publicly known numbers a prime no. q & an integer a that is a primitive root of q.
- Suppose the users A & B wish to exchange a key. User A selects a random integer $X_A < q$ & computes $Y_A = \alpha^{X_A} \bmod q$. Similarly User B independently selects a random integer $X_B < q$ & computes $Y_B = \alpha^{X_B} \bmod q$.

Each side keeps the value X private & makes the value Y available publicly to the other side.

User A computes the key K as $K = Y_B^{X_A} \bmod q$.
User B computes the key K as $K = Y_A^{X_B} \bmod q$.
these two calculation produce identical results.

Eg:- 1) Users Alice & Bob who wish to swap keys.
2) Agree on prime $q = 353$ & $a = 3$.
3) Select random secret keys :
- A chooses $X_A = 97$, B choose $X_B = 233$.
4) Compute public keys.
- $Y_A = 3^{97} \bmod 353 = 40$ (Alice)
- $Y_B = 3^{233} \bmod 353 = 248$ (Bob)
5) Compute shared session key as
$K_{AB} = Y_B^{X_A} \bmod 353 = 248^{97} = 160$ (Alice).
$K_{AB} = Y_A^{X_B} \bmod 353 = 40^{233} = 160$ (Bob).

Scanned with CamScanner

Global Public Elements

| | |
|---|---|
| q | Prime number |
| a | $a < q$ & $a$ a primitive root of $q$. |

User A Key Generation

Select private $X_A$ $\quad X_A < q$

Calculate public $Y_A$ $\quad Y_A = \alpha^{X_A} \bmod q$.

Calculation of Select key by User A

$K = (Y_B)^{X_A} \bmod q$.

Calculation of Secret key by User B.

$K = (Y_A)^{X_B} \bmod q$.

Figure :- Diffie Hellman Process.

Conclusion :-

We have studied & implemented Diffie Hellman key exchange algorithm.

Thank you

Scanned with CamScanner

Name: Umang Patel

Roll no: 2193271

Is lab – Assignment 04 - Deffie Hellman Algorithm

Code:

```
from random import randint
if __name__ == '__main__':

    P = 23
    G = 9

    print('The Value of P is :%d'%(P))
    print('The Value of G is :%d'%(G))

    a = 4
    print('The Private Key a for Alice is :%d'%(a))

    x = int(pow(G,a,P))

    b = 3
    print('The Private Key b for Bob is :%d'%(b))

    y = int(pow(G,b,P))

    ka = int(pow(y,a,P))

    kb = int(pow(x,b,P))

    print('Secret key for the Alice is : %d'%(ka))
    print('Secret Key for the Bob is : %d'%(kb))
```

Scanned with CamScanner

Name - Umang Patel.
Subject - Information Security
Roll No. - 2193271.
Class - LY-IS-3

## Assignment - 06

**Aim :-**

Install and Configure Kali linux for pentesting.

**Objective :-**

1) To learn the basics of Kali - Linux.
2) To Configure Kali Linux for penetration testing.

**Theory :-**

Kali Linux is an open source, Debian-based Linux distribution aimed at advanced Penetration Testing & Security Auditing. It does this by providing common ~~task~~ tools, configuration & automations which allow the user to focus on the task that needs to be completed, not the surrounding activity. It is geared towards various information security tasks such as Penetration testing, Security Research, Computer forensics & Reverse Engeneering.

Scanned with CamScanner

Kali Linux is a multi-platform solution, accessible & freely available to information Security professionals & hobbyists.

## Kali Linux Installation.

Different ways of installing Kali Linux.

1) Bootable device.
2) Hard disk installation.
3) Dual Boot.
4) Virtual - Box.

### Kali Linux Using Bootable USB.

Step 1: Download Kali Linux ISO. Download the ISO image file of Kali Linux from its official website. www.kali.org.

Step 2: You should download any bootable Image file Procesing tool, which allows you to create a bootable USB drive

Step 3: Play your USB drive into your Windows PC & remember the drive name associated with it ("G:")

Step 4: Launch Rufus & choose the Kali Linux ISO file. that you want to copy on the USB drive. Match the name of the device selected, which corresponds to the name of the USB ~~drive~~ device you inserted.

Scanned with CamScanner

Step 5 :- Click the Start button to overwrite the contents of USB drive.

Step 6 :- Once the status bar is completed 100%, close & safely rename USB drive from the PC. You can now use your USB drive to boot Kali Linux.

Step 7 :- Insert your bootable USB drive to your PC & start your PC. Press "Esc" on the first prompt to enter into the Startup Menu.

Step 8 :- Press F9 to enter into Boot Device option. It will redirect you to the Boot Manager. These option may vary from device to device

Step 9 :- Select USB hard drive to boot from your bootable USB drive. It will redirect you to the Kali Linux Live Boot Menu.

Step 10 :- Select Live System & press enter to boot Kali Linux.

You can now use Kali Linux on the same hardware without disturbing your host operating system.

Conclusion :- We have installed Kali Linux on Windows using bootable USB device.

Scanned with CamScanner

Name: Umang Patel

Roll no: 2193271

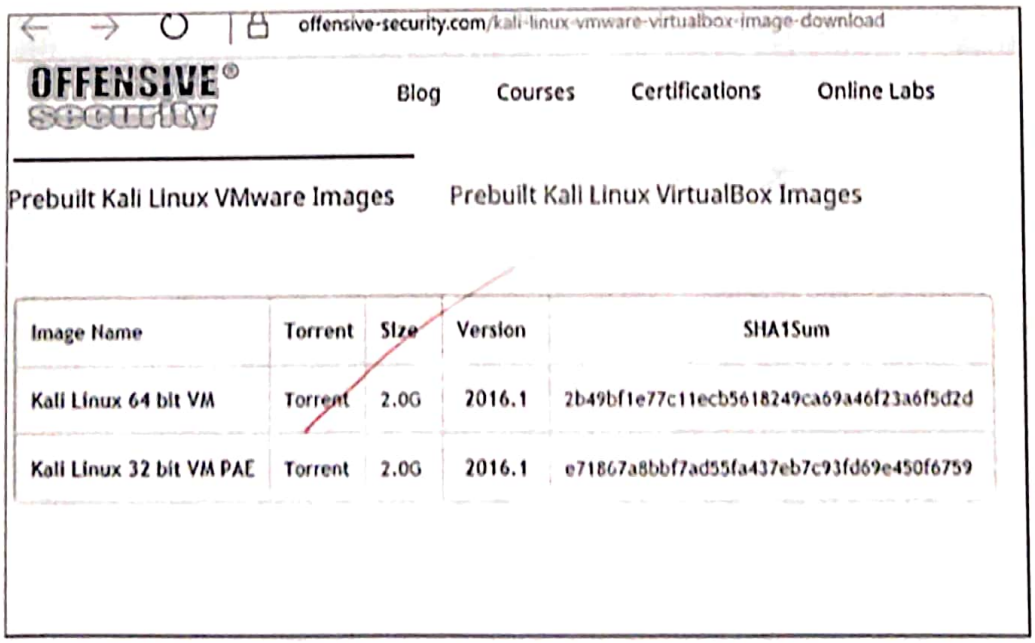
Is lab – Assignment 05 Kali Linux

Implementation:-

official webpage of kali Linux is https://www.kali.org. Backtrack was the old version of Kali Linux distribution. The latest release is Kali 2016.1 and it is updated very often.


Install Kali Linux

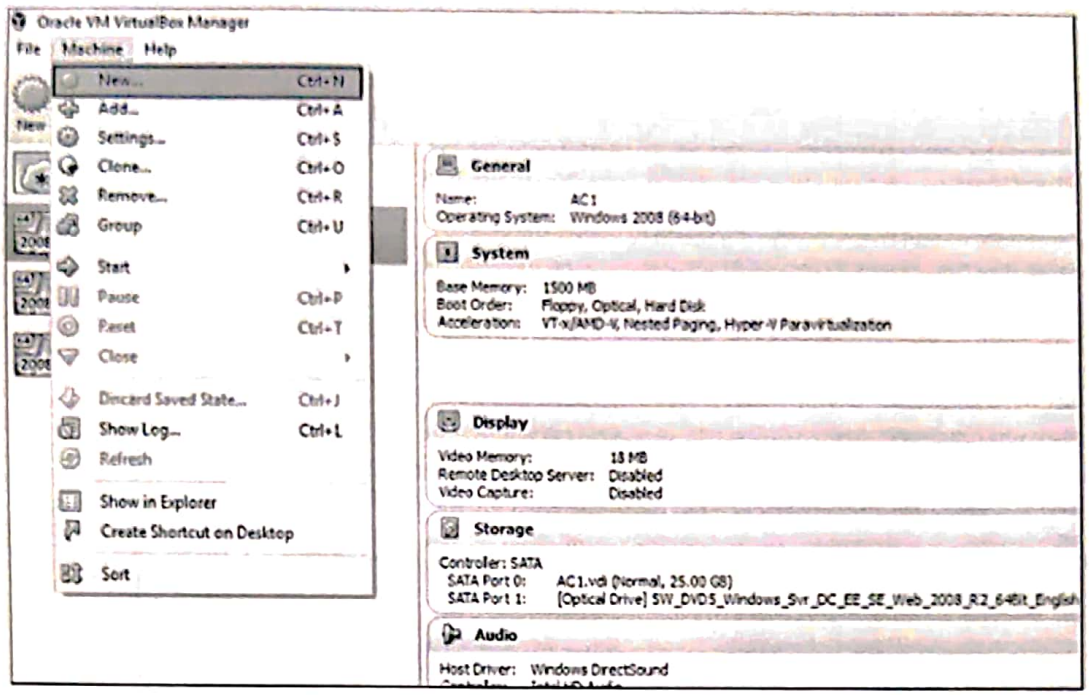
Step 1) Download the Kali Linux package from its official website: https://www.kali.org/downloads/



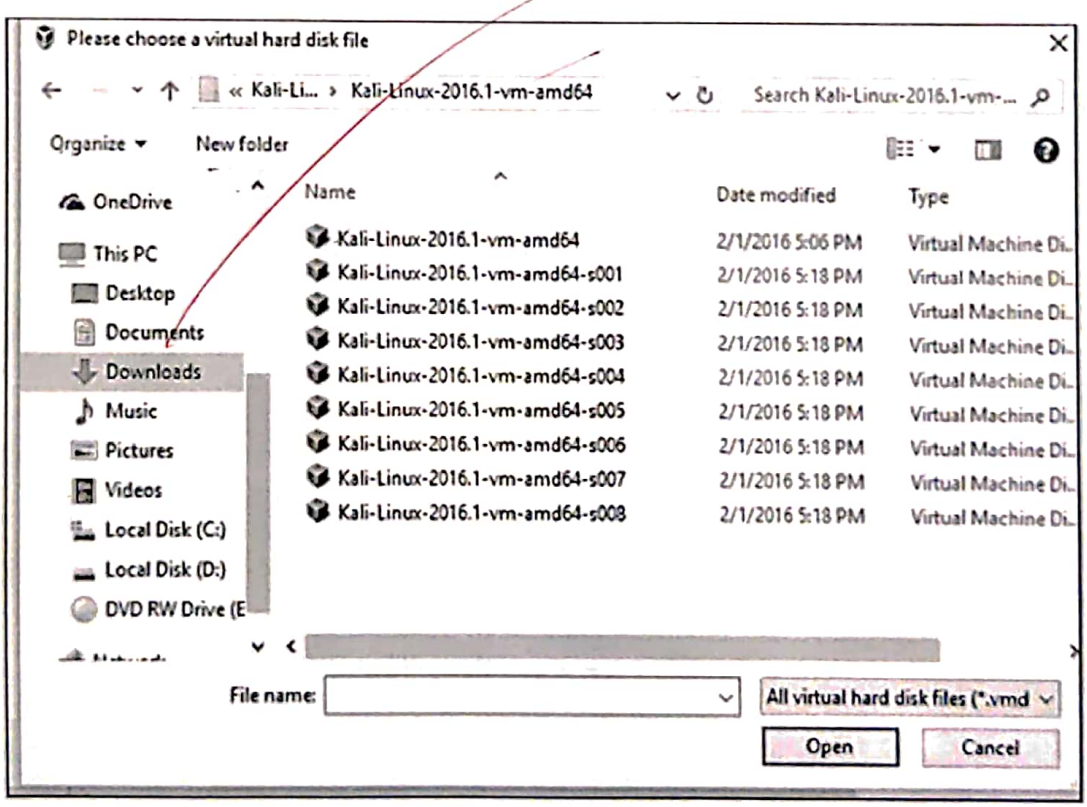

| Image Name | Torrent | Size | Version | SHA1Sum |
|---|---|---|---|---|
| Kali Linux 64 bit VM | Torrent | 2.0G | 2016.1 | 2b49bf1e77c11ecb5618249ca69a46f23a6f5d2d |
| Kali Linux 32 bit VM PAE | Torrent | 2.0G | 2016.1 | e71867a8bbf7ad55fa437eb7c93fd69e450f6759 |

Scanned with CamScanner

Step 2) Click **VirtualBox → New** as shown in the following screenshot.

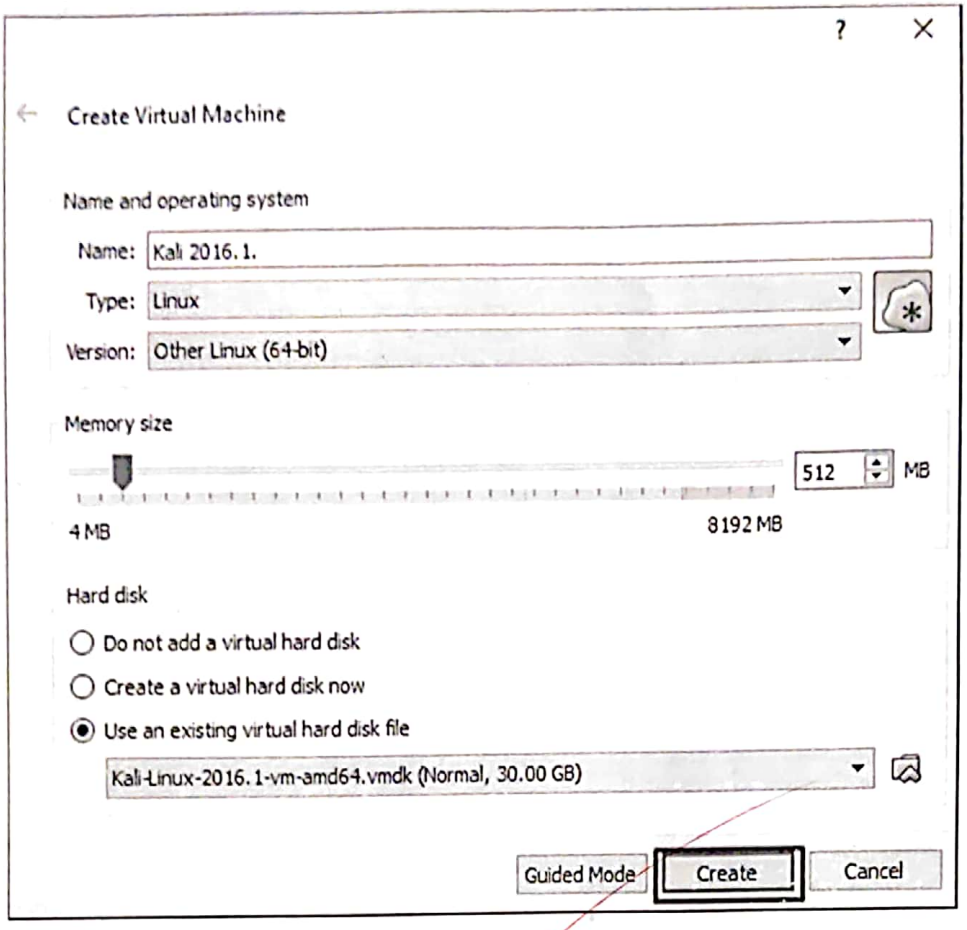



Step 3) Choose the right **virtual hard disk file** and click **Open**.

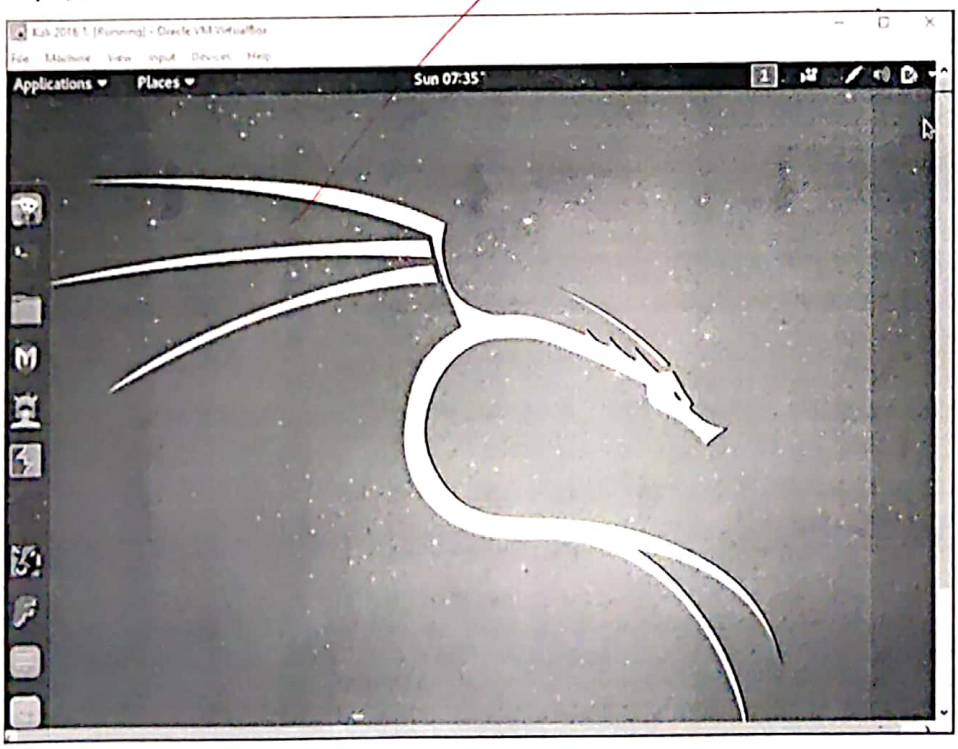



Scanned with CamScanner

Step 4) The following screenshot pops up. Click the **Create** button.




Step 5) Start Kali OS. The default username is **root** and the password is **toor**.




Scanned with CamScanner