**Name :- Harsh Shah**

**CLass:- TE Computer**

**ERP :-67**

**Subject :-LP2(IS) (AES)**

**Code:-**

import hashlib  
from base64 import b64decode, b64encode  
  
from Crypto import Random  
from Crypto.Cipher import AES  
  
  
class AESCipher(object):  
 def \_\_init\_\_(self, key):  
 self.block\_size = AES.block\_size  
 self.key = hashlib.sha256(key.encode()).digest()  
  
 def encrypt(self, plain\_text):  
 plain\_text = self.\_\_pad(plain\_text)  
 iv = Random.new().read(self.block\_size)  
 cipher = AES.new(self.key, AES.MODE\_CBC, iv)  
 encrypted\_text = cipher.encrypt(plain\_text.encode())  
 return b64encode(iv + encrypted\_text).decode("utf-8")  
  
 def decrypt(self, encrypted\_text):  
 encrypted\_text = b64decode(encrypted\_text)  
 iv = encrypted\_text[:self.block\_size]  
 cipher = AES.new(self.key, AES.MODE\_CBC, iv)  
 plain\_text = cipher.decrypt(encrypted\_text[self.block\_size:]).decode("utf-8")  
 return self.\_\_unpad(plain\_text)  
  
 def \_\_pad(self, plain\_text):  
 number\_of\_bytes\_to\_pad = self.block\_size - len(plain\_text) % self.block\_size  
 ascii\_string = chr(number\_of\_bytes\_to\_pad)  
 padding\_str = number\_of\_bytes\_to\_pad \* ascii\_string  
 padded\_plain\_text = plain\_text + padding\_str  
 return padded\_plain\_text  
  
 @staticmethod  
 def \_\_unpad(plain\_text):  
 last\_character = plain\_text[len(plain\_text) - 1:]  
 return plain\_text[:-ord(last\_character)]  
  
  
key = input("Enter Key: ")  
aes = AESCipher(key)  
  
flag = 1  
while flag == 1:  
 print("/\*\*\*\*\*\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*\*\*\*\*\*\*/")  
 print("1. Encryption")  
 print("2. Decryption")  
 print("3. Exit ")  
 choice = int(input("Enter your choice : "))  
  
 if choice == 1:  
 message = input("Enter message to encrypt: ")  
 encryptedMessage = aes.encrypt(message)  
 print("Encrypted Message:", encryptedMessage)  
  
 elif choice == 2:  
 message = input("Enter message to decrypt: ")  
 decryptedMessage = aes.decrypt(message)  
 print("Decrypted Message:", decryptedMessage)  
 elif choice == 3:  
 print("Exit")  
 flag = 0  
 else:  
 print("Wrong Choice,Please Choose Another Option.")

**Output:-**

Enter Key: AISSMSIOIT

/\*\*\*\*\*\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

1. Encryption

2. Decryption

3. Exit

Enter your choice : 1

Enter message to encrypt: Its Lonewolf aka Harsh

Encrypted Message: mqVOvzwWn33k5g5K/iht4GhatMGOmDPlgoSQWW4YerGuKMB3/zomXaq9HJvk4rVS

/\*\*\*\*\*\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

1. Encryption

2. Decryption

3. Exit

Enter your choice : 2

Enter message to decrypt: mqVOvzwWn33k5g5K/iht4GhatMGOmDPlgoSQWW4YerGuKMB3/zomXaq9HJvk4rVS

Decrypted Message: Its Lonewolf aka Harsh

/\*\*\*\*\*\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

1. Encryption

2. Decryption

3. Exit

Enter your choice : 5

Wrong Choice,Please Choose Another Option.

/\*\*\*\*\*\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

1. Encryption

2. Decryption

3. Exit

Enter your choice : 3

Exit

Process finished with exit code 0