Write a C program that contains a string (char pointer) with a value \Hello World’. The program should XOR each character in this string with 0 and displays the result.

string=[]  
string.extend("Hello World")  
str1=""  
for i in string:  
 str1+=chr(ord(i)^0)  
print(str1)

Write a C program that contains a string (char pointer) with a value \Hello World’. The program should AND or and XOR each character in thisstring with 127 and display the result

string=[]  
string.extend("Hello World")  
str1=""  
str2=""  
for i in string:  
 str1+=chr(ord(i)&127)  
print(str1)  
  
for i in string:  
 str2+=chr(ord(i)^127)  
print(str1)

Write a Java program to perform encryption and decryption using the following algorithms:

1. Ceaser Cipher
2. alfavit = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz'  
   print(len(alfavit))  
   text=input("Text: ")  
   key=int(input("Key: "))  
   mode=input("Choose mode: ")  
   itog=""  
   if mode=="E":  
    for i in range(len(text)):  
    if (alfavit.find(text[i])+key)<=51:  
    itog+=alfavit[alfavit.index(text[i])+key]  
    else:  
    itog += alfavit[(alfavit.index(text[i]) + key)-52]  
   else:  
    for i in range(len(text)):  
    if (alfavit.find(text[i])-key)>=0:  
    itog+=alfavit[alfavit.index(text[i])-key]  
    else:  
    itog += alfavit[(alfavit.index(text[i]) - key)+26]  
     
     
     
   print(itog)
3. Substitution Cipher
4. alphabet = "abcdefghijklmnopqrstuvwxyz"  
   cipher = "rqponmlkjihgfedcbazyxwvuts"  
   text = input("Enter text: ")  
   encrypt = ""  
   for i in text:  
    encrypt += cipher[alphabet.find(i)]  
   print(encrypt)

c) Hill Cipher

from re import findall  
MatrixKey = [[11,8],[4,3]]  
iMatrixKey = [  
 [MatrixKey[1][1],-MatrixKey[0][1]],  
 [-MatrixKey[1][0],MatrixKey[0][0]]  
]  
alpha = tuple("ABCDEFGHIJKLMNOPQRSTUVWXYZ")  
cryptMode = input("Chose Mode: ").upper()  
  
startMessage = input("Write your message: ").upper()  
for symbol in startMessage:  
 if symbol not in [chr(x) for x in range(65,91)]:  
 startMessage = startMessage.replace(symbol,'')  
while len(startMessage) % 2 != 0: startMessage += 'Z'  
def regular(text):  
 template = r"[A-Z]{"+str(2)+"}"  
 return findall(template, text)  
def encryptDecrypt(message, matrix, summ = 0, final = ""):  
 for double in range(len(message)):  
 for string in range(2):  
 for column in range(2):  
 summ += matrix[string][column] \* alpha.index(message[double][column])  
 final += alpha[(summ)%26]; summ = 0  
 return final  
if cryptMode == 'E': finalMessage = encryptDecrypt(regular(startMessage), MatrixKey)  
else: finalMessage = encryptDecrypt(regular(startMessage), iMatrixKey)  
print("Final message:",finalMessage)