Lab 3: Classical Cryptography

Thực hành môn Nhập môn bảo đảm và an ninh thông tin

Nhóm: 8

Lớp: IE105.O12.CNCL.1

Danh sách thành viên:

MSSV	Họ tên	Đánh giá
21522405	Nguyễn Thị Nhàn	Tốt
21522411	Trần Trọng Nhân	Tốt
21522557	Trần Thanh Sơn	Tốt
21522677	Nguyễn Trí Tín	Tốt
21522777	Trương Quang Tường	Tốt

Link app của nhóm: https://21522557-ie105lab3.streamlit.app/

MỤC LỤC

Task 1: Let's begin with a straightforward task that does not use any cipher algorithm. Try to solve the following codes:

1. We need to find the code to open the lock in figure 4. The lock has 3 digit pin which satisfies 5 conditions (hints) that are given . Can you crack this code? If it's possible, explain how.

Dựa vào các gợi ý, chúng ta có thể phân tích như sau:

- Từ gợi ý đầu tiên, chúng ta biết rằng 6, 8, hoặc 2 là một trong những số đúng và đúng vị trí. Từ gợi ý thứ hai, chúng ta biết rằng 6 không phải là số đúng, vì vậy nó phải là 8 hoặc 2. Số đúng khác có thể là 1 hoặc 4.
- Từ gợi ý thứ ba, chúng ta biết rằng 2 và 0 đều là những số có trong mã số, nhưng không đúng vị trí. Vì vậy, 2 phải ở vị trí thứ hai hoặc thứ ba, và 0 phải ở vị trí thứ nhất hoặc thứ ba.
- Từ gợi ý thứ tư, chúng ta biết rằng 7, 3, và 8 không có trong mã số. Vì vậy, những số còn lại có thể là 0, 1, 4, 5, và 9. Kết hợp với gợi ý đầu tiên, suy ra số 2 là số đúng và nằm ở vị trí thứ ba.

- Từ gợi ý thứ năm, chúng ta biết rằng 7, 8, hoặc 0 là một trong những số có trong mã số, nhưng không đúng vị trí. Vì 7 và 8 không có trong mã số, nên nó phải là 0. Và vì 0 không phải là số ở vị trí thứ ba, nên nó phải ở vị trí thứ nhất hoặc thứ hai. Kết hợp với gợi ý thứ ba, suy ra số 0 nằm ở vị trí thứ nhất.
- Xét lại gợi ý thứ 2, chúng ta biết rằng 1 hoặc 4 là số đúng nhưng sai vị trí. Vì 1 không thể nằm ở vị trí thứ nhất hoặc thứ ba do trong mã số đã có 0 và 2 ở hai vị trí đó nên 4 là số đúng và nằm ở vị trí thứ hai.

Vậy mã khóa chính xác là 042.

2. Find the corresponding encoding for the numbers 1 to 9 according to the clues provided in table 1.2.1

- Each symbol in the set (△◁ ▷◯ ♡ ♠ ◊ ♣ ●) unique encoding for one of the number from 1 to 9.
- The rightmost column is the sum of the numbers in each row.
- The bottom row is the sum of the numbers in each column.
- Each ? can represent any one-digit or two-digit number and could be same or different from each other.

Table 1. Find the corresponding encoding for each number.

△	△	◁	◯	?
♡	♡	♠	♡	◊◊
?	?	◁	♣	●●
?	♡	♠	♡	●▷
●♡	◊◊	●●	●◊	

Dựa vào các gợi ý ta có hệ phương trình sau:

3*♡ + ♠ = ◊◊

2*◁ + 2*♠ = ●●

◯ + 2*♡ + ♣ = ●◊

Bằng một vài dòng code ta đã giải được phương trình trên với 2 nghiệm.

```
In [128]: def solve_equations():
              for a in range(1, 10):
                  for b in range(1, 10):
                      for c in range(1, 10):
                          for d in range(1, 10):
                              for e in range(1, 10):
                                  for f in range(1, 10):
                                      for g in range(1, 10):
                                          # Kiểm tra các chữ số phải khác nhau
                                          if len(set([a, b, c, d, e, f, g])) == 7:
                                              # Điều kiện của phương trình 1
                                              if 3 * a + b == c*10 +c:
                                                  # Điều kiện của phương trình 2
                                                  if 2 * d + 2 * b == e*10+e:
                                                      # Điều kiện của phương trình 3
                                                      if f + 2 * a + g == e*10 + c:
                                                          # In ra giá trị a, b, c, d, e, f
                                                          print(f"♡={a}, ♠={b}, ◊={c}, ◁=

          # Gọi hàm để giải hệ phương trình
          solve_equations()

          ♡=9, ♠=6, ◊=3, ◁=5, ●=2, ◯=1, ♣=4
          ♡=9, ♠=6, ◊=3, ◁=5, ●=2, ◯=4, ♣=1
```

Ta chắc chắn rằng ♡ = 9, ♠ = 6, ◊ = 3, ◁ = 5, ● = 2

Còn 2 trường hợp:

◯ = 1, ♣ = 4

◯ = 4, ♣ = 1

⇨ Các kí tự còn lại là ▷ và △ thuộc {7, 8}

Gọi dấu "?" ở các ô có vị trí (3,1), (3,2), (4,1) lần lượt là x, y, z

$$\begin{cases} x + y + ♣ + 5 = 22 \\ x + z + △ + 9 = 29 \\ y + △ + 18 = 33 \\ z + 18 + 6 = 20 + ▷ \end{cases} \Rightarrow \begin{cases} x + y + ♣ + 5 = 22 \\ y + 9 - x - z = 4 \\ z + 18 + 6 = 20 + ▷ \end{cases}$$

Xét trường hợp ▷ = 7 thì △ = 8, z = 3:

+ Giả sử ♣ = 1 thì x = 9, y = 7. Thế vào phương trình y + △ + 18 = 33 thì △ = 8 (TM).

=> ◯ = 4

+ Giả sử ♣ = 4 thì x = 7.5, y = 5.5 (Loại vì x, y phải là số nguyên).

Xét trường hợp ▷ = 8 thì △ = 7, z = 4:

+ Giả sử ♣ = 1 thì x = 8.5, y = 7.5 (Loại vì x, y phải là số nguyên).

+ Giả sử ♣ = 4 thì x = 7, y = 6. Thế vào phương trình y + △ + 18 = 33 thì △ = 9 (Loại).

Vậy ♡ = 9, ♠ = 6, ◊ = 3, ◁ = 5, ● = 2, ◯ = 4, ♣ = 1, ▷ = 7, △ = 8 và ta có bảng giải mã sau:

△	△	◁	◯	●◁
♡	♡	♠	♡	◊◊
♡	▷	◁	♣	●●
◊	♡	♠	♡	●▷
●♡	◊◊	●●	●◊	

Task 2: In this task, you must create an application in your chosen programming language that performs encryption and decryption using the **Caesar** cipher. The application should meet the following criteria:

- Enable user input for a key and either plaintext for encryption or ciphertext for decryption.

Provide the ability to perform a brute-force attack by trying all possible keys to decrypt a given ciphertext without knowing the key.

- To validate your program, test it with a message of at least 100 words and compare the results with other cryptography online tool, such as dcode (https://www.dcode.fr), CrypTool 2 online. Additionally, use your program to crack the following ciphertext:

Mfwzpn Rzwfpfrn bfx gtws ns Pdtyt ns 1949 fsi stb qnajx sjfw Ytpdt. Mj nx ymj fzymtw tk rfsd stajqx fx bjqq fx xmtwy xytwnjx fsi sts-knhynts. Mnx btwpx nshqzij Stwbjlnfs Btti, Ymj Bnsi-Zu Gnwi Hmwtsnhqj, Pfkpf ts ymj Xmtwj, Fkyjw Ifwp fsi Bmfy N Yfqp Fgtzy Bmjs N Yfqp Fgtzy Wzssnsl. Mnx btwp mfx gjjs ywfsxqfyji nsyt rtwj ymfs ktwyd qfslzfljx, fsi ymj rtxy wjhjsy tk mnx rfsd nsyjwsfyntsfq mtstzwx nx ymj Ojwzxfqjr Uwnej, bmtxj uwjantzx wjhnunjsyx nshqzij O.R. Htjyejj, Rnqfs Pzsijwf, fsi A.X. Sfnufzq.

Do you find any special concerning the key used to encrypt this ciphertext?

Hàm caesar_cipher

```
def caesar_cipher(text, key, decrypt=False):
    result = ""
    for char in text:
        if char.isalpha():
            shift = key % 26
            if char.isupper():
                result += chr((ord(char) - shift - 65) % 26 + 65) if decrypt else chr((ord(char) + shift - 65) % 26 + 65)
            else:
                result += chr((ord(char) - shift - 97) % 26 + 97) if decrypt else chr((ord(char) + shift - 97) % 26 + 97)
        else:
            result += char
    return result
```

- text: Chuỗi cần được mã hóa hoặc giải mã.
- key: Khóa được sử dụng cho mã hóa hoặc giải mã. Là một số từ 0 đến 25.
- decrypt: Một biến boolean quyết định xem hàm sẽ thực hiện mã hóa hay giải mã. Mặc định là False để thực hiện mã hóa.

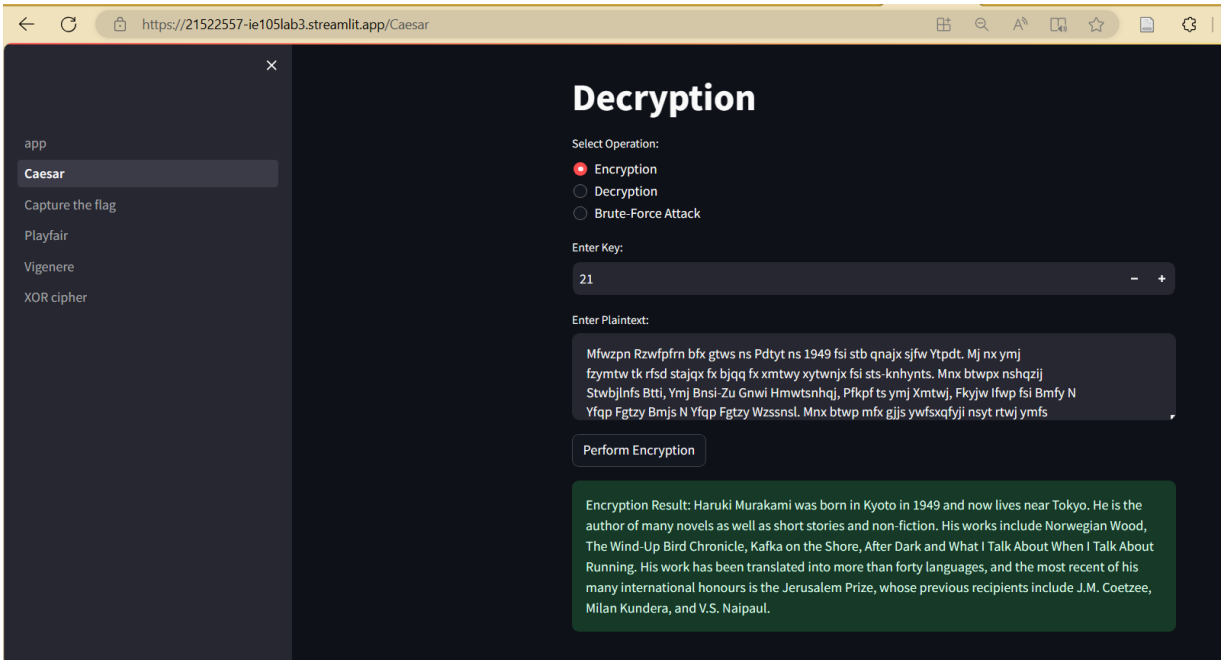
Vòng lặp duyệt qua từng ký tự trong chuỗi đầu vào (text):

- Nếu ký tự là một chữ cái (isalpha()), hàm sẽ thực hiện quá trình mã hóa hoặc giải mã dựa trên giá trị của decrypt.
- shift: Số lượng chuyển dịch trong bảng chữ cái.
- Nếu ký tự là chữ cái in hoa (isupper()), hàm sẽ thực hiện phép toán mã hóa hoặc giải mã cho chữ cái in hoa.
- Nếu ký tự là chữ cái thường, hàm sẽ thực hiện phép toán tương ứng cho chữ cái thường.
- Nếu ký tự không phải là chữ cái, giữ nguyên ký tự đó.
- Kết quả cuối cùng là chuỗi đã được mã hóa hoặc giải mã.

Hàm brute_force_decrypt

```
def brute_force_decrypt(ciphertext):
    results = {}
    for key in range(26):
        decrypted_text = caesar_cipher(ciphertext, key, decrypt=True)
        results[key] = decrypted_text
    return results
```

- ciphertext: Chuỗi đã được mã hóa cần được giải mã bằng phương pháp tấn công brute-force.
- Hàm này thực hiện một vòng lặp qua mọi khóa từ 0 đến 25.
- Mỗi lần vòng lặp, nó sử dụng hàm caesar_cipher để giải mã với khóa tương ứng.
- Kết quả được lưu vào một từ điển (results) với khóa là giá trị của khóa thử và giá trị là văn bản đã giải mã.

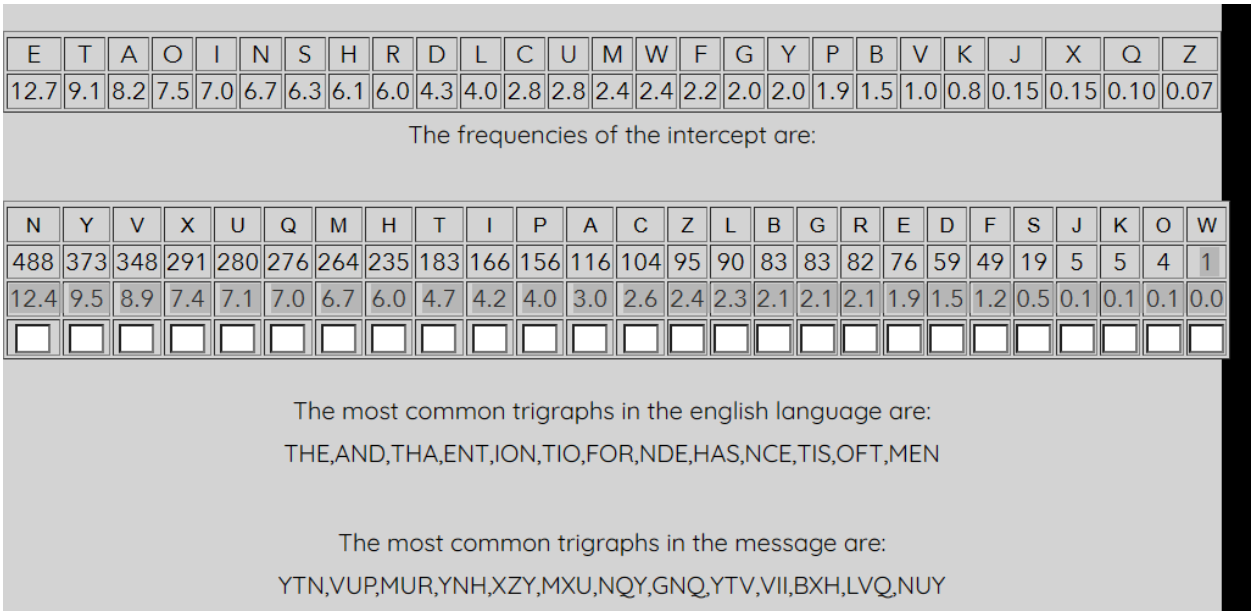


Task 3: Click **here** to download the ciphertext file 2.

Your job is to find out the original text using frequency analysis. It is known that the original text is an English article.

Describe how to find the plain-text in detail (step-by-step).

- Giải mã văn bản bằng cách sử dụng frequency analysis:



E	T	A	O	I	N	S	H	R	D	L	C	U	M	W	F	G	Y	P	B	V	K	J	X	Q	Z
12.7	9.1	8.2	7.5	7.0	6.7	6.3	6.1	6.0	4.3	4.0	2.8	2.8	2.4	2.4	2.2	2.0	2.0	1.9	1.5	1.0	0.8	0.15	0.15	0.10	0.07

The frequencies of the intercept are:

N	Y	V	X	U	Q	M	H	T	I	P	A	C	Z	L	B	G	R	E	D	F	S	J	K	O	W
488	373	348	291	280	276	264	235	183	166	156	116	104	95	90	83	83	82	76	59	49	19	5	5	4	1
12.4	9.5	8.9	7.4	7.1	7.0	6.7	6.0	4.7	4.2	4.0	3.0	2.6	2.4	2.3	2.1	2.1	2.1	1.9	1.5	1.2	0.5	0.1	0.1	0.1	0.0
☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

The most common trigraphs in the english language are:

THE,AND,THA,ENT,ION,TIO,FOR,NDE,HAS,NCE,TIS,OFT,MEN

The most common trigraphs in the message are:

YTN,VUP,MUR,YNH,XZY,MXU,NQY,GNQ,YTV,VII,BXH,LVQ,NUY

- Theo thông số trên ta thử thay 'ytn' thành 'the', 'vup' thành 'and'. Tiếp theo ta có thể thay 'x' bằng 'o' dựa theo chỉ số định lượng. Ta thay thế 'ytnvupx' thành 'theando' ở một phần của văn bản:

THE OqaAhq TzhN ON qzNDAd lHmaH qEEcq AgOzT hmrHT AbTEh THmq iONr qThANrE AlAhDq Thme THE gArrEh bEEiq imsE A NONArENAhmAN TOO

THE AlAhDq hAaE lAq gOOsENDED gd THE DEcmqE Ob HAhfEd lEmNqTEmN AT mTq OzTqET AND THE AeeAhENT mceiOqmON Ob Hmq bmic aOceANd AT THE END AND mT lAq qHAeED gd THE EcEhrENaE Ob cETOO TmcEq ze giAasrOlN eOimTmaq AhcaANDd AaTmfmqc AND A NATmONAi aONfEhqATmON Aq ghmEb AND cAD Aq A bEfEh DhEAc AgOzT lHETHEh THEhE OzrHT TO gE A ehEqmDENT lmNbhEd THE qEAqON DmDNT ozqT qEEc EkThA iONr mT lAq EkThA iONr gEaAzqE THE OqaAhq lEhE cOfED TO THE bmhqT lEEsEND mN cAhaH TO AfOmD aONbimaTmNr lmTH THE aiOqmNr aEhEcONd Ob THE lmNTEh Oidcemaq THANsq edEONraHANr

- Ta thấy cụm từ "NATmONAi aONfEhqATmON" khá giống với "NATIONAL CONVERSATION", "THEhE OzrHT TO gE" khá giống với "THERE OUGHT TO BE". Ta thử thay thế 'miafhqzrg' thành 'ilcvrsugb' ở một phần của văn bản trên:

THE OSCARS TURN ON SUNDAd lHICH SEEcS ABOUT RIGHT AbTER THIS LO

NG STRANGE AlARDS TRIe THE BAGGER bEELS LIsE A NONAGENARIAN TO O

THE AlARDS RACE lAS BOOsENDED Bd THE DEcISE Ob HARVEd lEINSTEIN A T ITS OUTSET AND THE AeeARENT IceLOSION Ob HIS bILc COceANd AT THE E ND AND IT lAS SHAeED Bd THE EcERGENCE Ob cETOO TIcES Ue BLACsGOlN e OLITICS ARcCANDd ACTIVISc AND A NATIONAL CONVERSATION AS BRIEb AND cAD AS A bEVER DREAc ABOUT lHETHER THERE OUGHT TO BE A eRESI DENT lINbREd THE SEASON DIDNT oUST SEEc EkTRA LONG IT lAS EkTRA LO NG BECAUSE THE OSCARS lERE cOVED TO THE bIRST lEEsEND IN cARCH TO AVOID CONbLICTING lITH THE CLOSING CEREcONd Ob THE lINTER OLdceICS THANsS edEONGCHANG

- Có vẻ như đoạn văn đang nói về giải Oscar của một người nào đó.

=> AlARDS = AWARDS, CONbLICTING = CONFLICTING, EkTRA = EXTRA, eRESIDENT= PRESIDENT, THANsS = THANKS, SUNDAd = SUNDAY

Ta thử thay thế 'lbke' thành 'wfxp' ở một phần của văn bản trên:

THE OSCARS TURN ON SUNDAY WHICH SEEcS ABOUT RIGHT AFTER THIS L ONG STRANGE AWARDS TRIP THE BAGGER FEELS LIKE A NONAGENARIAN TOO

THE AWARDS RACE WAS BOOKENDED BY THE DEcISE OF HARVEY WEINSTE IN AT ITS OUTSET AND THE APPARENT IcPLOSION OF HIS FILc COcPANY AT T HE END AND IT WAS SHAPED BY THE EcERGENCE OF cETOO TIcES UP BLAC KGOWN POLITICS ARcCANDY ACTIVISc AND A NATIONAL CONVERSATION A S BRIEF AND cAD AS A FEVER DREAc ABOUT WHETHER THERE OUGHT TO B E A PRESIDENT WINFREY THE SEASON DIDNT oUST SEEc EXTRA LONG IT W AS EXTRA LONG BECAUSE THE OSCARS WERE cOVED TO THE FIRST WEEKE ND IN cARCH TO AVOID CONFLICTING WITH THE CLOSING CEREcONY OF T HE WINTER OLYcPICS THANKS PYEONGCHANG

- Đoạn văn có vẻ rõ ràng hơn, ta tiếp tục thay thế FILc COcPANY = FILM COMPA NY => Thay 'c' thành 'm' ta được:

THE AWARDS RACE WAS BOOKENDED BY THE DEMISE OF HARVEY WEINS TEIN AT ITS OUTSET AND THE APPARENT IMPLOSION OF HIS FILM COMPAN Y AT THE END AND IT WAS SHAPED BY THE EMERGENCE OF METOO TIMES UP BLACKGOWN POLITICS ARMCANDY ACTIVISM AND A NATIONAL CONV ERSATION AS BRIEF AND MAD AS A FEVER DREAM ABOUT WHETHER THE

RE OUGHT TO BE A PRESIDENT WINFREY THE SEASON DIDNT oUST SEEM E
XTRA LONG IT WAS EXTRA LONG BECAUSE THE OSCARS WERE MOVED TO
THE FIRST WEEKEND IN MARCH TO AVOID CONFLICTING WITH THE CLOSI
NG CEREMONY OF THE WINTER OLYMPICS THANKS
PYEONGCHANG

ONE BIG jUESTION SURROUNDING THIS YEARS ACADEMY AWARDS IS HO
W OR IF THE CEREMONY WILL ADDRESS METOO ESPECIALLY AFTER THE G
OLDEN GLOBES WHICH BECAME A oUBILANT COMINGOUT PARTY FOR TIM
ES UP THE MOVEMENT SPEARHEADED BY POWERFUL HOLLYWOOD WOME
N WHO HELPED RAISE MILLIONS OF DOLLARS TO FIGHT SEXUAL HARASS
MENT AROUND THE COUNTRY

- Sau khi chuyển đổi thì văn bản gần như đã hoàn thiện với số lượng key hiện tại là 2 3 chỉ có một số từ chưa được chuyển đổi như 'oUST', 'PRIwE' ,'jUESTION'. Dựa theo ngữ cảnh ta chuyển đổi 'owj' thành 'jmq'.

Như vậy với key "ytnvupxmiafhqzrglbkesdcowj" = "theandoilcvrsugbwfxpkymjmq" ta được:

THE OSCARS TURN ON SUNDAY WHICH SEEMS ABOUT RIGHT AFTER THIS
LONG STRANGE
AWARDS TRIP THE BAGGER FEELS LIKE A NONAGENARIAN TOO

THE AWARDS RACE WAS BOOKENDED BY THE DEMISE OF HARVEY WEINS
TEIN AT ITS OUTSET
AND THE APPARENT IMPLOSION OF HIS FILM COMPANY AT THE END AND I
T WAS SHAPED BY
THE EMERGENCE OF METOO TIMES UP BLACKGOWN POLITICS ARMCANDY
ACTIVISM AND
A NATIONAL CONVERSATION AS BRIEF AND MAD AS A FEVER DREAM ABO
UT WHETHER THERE
OUGHT TO BE A PRESIDENT WINFREY THE SEASON DIDNT JUST SEEM EXT
RA LONG IT WAS
EXTRA LONG BECAUSE THE OSCARS WERE MOVED TO THE FIRST WEEKEN
D IN MARCH TO
AVOID CONFLICTING WITH THE CLOSING CEREMONY OF THE WINTER OLY
MPICS THANKS
PYEONGCHANG

ONE BIG QUESTION SURROUNDING THIS YEARS ACADEMY AWARDS IS HO
W OR IF THE
CEREMONY WILL ADDRESS METOO ESPECIALLY AFTER THE GOLDEN GLO
BES WHICH BECAME
A JUBILANT COMINGOUT PARTY FOR TIMES UP THE MOVEMENT SPEARHE
ADED BY
POWERFUL HOLLYWOOD WOMEN WHO HELPED RAISE MILLIONS OF DOLL
ARS TO FIGHT SEXUAL
HARASSMENT AROUND THE COUNTRY

SIGNALING THEIR SUPPORT GOLDEN GLOBES ATTENDEES SWATHED THE
MSELVES IN BLACK
SPORTED LAPEL PINS AND SOUNDED OFF ABOUT SEXIST POWER IMBALAN
CES FROM THE RED
CARPET AND THE STAGE ON THE AIR E WAS CALLED OUT ABOUT PAY INE
QUITY AFTER
ITS FORMER ANCHOR CATT SADLER QUIT ONCE SHE LEARNED THAT SHE
WAS MAKING FAR
LESS THAN A MALE COHOST AND DURING THE CEREMONY NATALIE PORT
MAN TOOK A BLUNT
AND SATISFYING DIG AT THE ALLMALE ROSTER OF NOMINATED DIRECTO
RS HOW COULD
THAT BE TOPPED

AS IT TURNS OUT AT LEAST IN TERMS OF THE OSCARS IT PROBABLY WON
T BE

WOMEN INVOLVED IN TIMES UP SAID THAT ALTHOUGH THE GLOBES SIGNI
FIED THE
INITIATIVES LAUNCH THEY NEVER INTENDED IT TO BE JUST AN AWARDS
SEASON
CAMPAIGN OR ONE THAT BECAME ASSOCIATED ONLY WITH REDCARPET
ACTIONS INSTEAD
A SPOKESWOMAN SAID THE GROUP IS WORKING BEHIND CLOSED DOORS
AND HAS SINCE
AMASSED MILLION FOR ITS LEGAL DEFENSE FUND WHICH AFTER THE GL
OBES WAS
FLOODED WITH THOUSANDS OF DONATIONS OF OR LESS FROM PEOPLE IN
SOME
COUNTRIES

NO CALL TO WEAR BLACK GOWNS WENT OUT IN ADVANCE OF THE OSCAR S THOUGH THE
MOVEMENT WILL ALMOST CERTAINLY BE REFERENCED BEFORE AND DUR ING THE CEREMONY
ESPECIALLY SINCE VOCAL METOO SUPPORTERS LIKE ASHLEY JUDD LAUR A DERN AND
NICOLE KIDMAN ARE SCHEDULED PRESENTERS

ANOTHER FEATURE OF THIS SEASON NO ONE REALLY KNOWS WHO IS GOI NG TO WIN BEST
PICTURE ARGUABLY THIS HAPPENS A LOT OF THE TIME INARGUABLY THE NAILBITER
NARRATIVE ONLY SERVES THE AWARDS HYPE MACHINE BUT OFTEN THE PEOPLE FORECASTING
THE RACE SOCALLED OSCAROLOGISTS CAN MAKE ONLY EDUCATED GUES SES

THE WAY THE ACADEMY TABULATES THE BIG WINNER DOESNT HELP IN E VERY OTHER
CATEGORY THE NOMINEE WITH THE MOST VOTES WINS BUT IN THE BEST PICTURE
CATEGORY VOTERS ARE ASKED TO LIST THEIR TOP MOVIES IN PREFERENT IAL ORDER IF A
MOVIE GETS MORE THAN PERCENT OF THE FIRSTPLACE VOTES IT WINS W HEN NO
MOVIE MANAGES THAT THE ONE WITH THE FEWEST FIRSTPLACE VOTES IS ELIMINATED AND
ITS VOTES ARE REDISTRIBUTED TO THE MOVIES THAT GARNERED THE ELI MINATED BALLOTS
SECONDPLACE VOTES AND THIS CONTINUES UNTIL A WINNER EMERGES

IT IS ALL TERRIBLY CONFUSING BUT APPARENTLY THE CONSENSUS FAVO RITE COMES OUT
AHEAD IN THE END THIS MEANS THAT ENDOFSEASON AWARDS CHATTER I NVARIABLY
INVOLVES TORTURED SPECULATION ABOUT WHICH FILM WOULD MOST LI KELY BE VOTERS
SECOND OR THIRD FAVORITE AND THEN EQUALLY TORTURED CONCLUSIO NS ABOUT WHICH
FILM MIGHT PREVAIL

IN IT WAS A TOSSUP BETWEEN BOYHOOD AND THE EVENTUAL WINNER BI RDMAN

IN WITH LOTS OF EXPERTS BETTING ON THE REVENANT OR THE BIG SHOR T THE
PRIME WENT TO SPOTLIGHT LAST YEAR NEARLY ALL THE FORECASTERS DECLARED LA
LA LAND THE PRESUMPTIVE WINNER AND FOR TWO AND A HALF MINUTES THEY WERE
CORRECT BEFORE AN ENVELOPE SNAFU WAS REVEALED AND THE RIGHTF UL WINNER
MOONLIGHT WAS CROWNED

THIS YEAR AWARDS WATCHERS ARE UNEQUALLY DIVIDED BETWEEN THR EE BILLBOARDS
OUTSIDE EBBING MISSOURI THE FAVORITE AND THE SHAPE OF WATER WH ICH IS
THE BAGGERS PREDICTION WITH A FEW FORECASTING A HAIL MARY WIN FOR GET OUT

BUT ALL OF THOSE FILMS HAVE HISTORICAL OSCARVOTING PATTERNS A GAINST THEM THE
SHAPE OF WATER HAS NOMINATIONS MORE THAN ANY OTHER FILM AND WAS ALSO
NAMED THE YEARS BEST BY THE PRODUCERS AND DIRECTORS GUILDS YE T IT WAS NOT
NOMINATED FOR A SCREEN ACTORS GUILD AWARD FOR BEST ENSEMBLE AND NO FILM HAS
WON BEST PICTURE WITHOUT PREVIOUSLY LANDING AT LEAST THE ACTO RS NOMINATION
SINCE BRAVEHEART IN THIS YEAR THE BEST ENSEMBLE SAG ENDED UP G OING TO
THREE BILLBOARDS WHICH IS SIGNIFICANT BECAUSE ACTORS MAKE UP T HE ACADEMYS
LARGEST BRANCH THAT FILM WHILE DIVISIVE ALSO WON THE BEST DRA MA GOLDEN GLOBE
AND THE BAFTA BUT ITS FILMMAKER MARTIN MCDONAGH WAS NOT NOM INATED FOR BEST
DIRECTOR AND APART FROM ARGO MOVIES THAT LAND BEST PICTURE WI THOUT ALSO
EARNING BEST DIRECTOR NOMINATIONS ARE FEW AND FAR BETWEEN

Task 4: Write an application with your own programming language to encrypt and decrypt a message using **Playfair cipher**. Your application should satisfy the following requirements:

- Allow you to input a key and a plain-text to encrypt or a cipher-text to decrypt using the given key.
- Display the Playfair matrix (5x5) corresponding with the given key.

1. Test your program with an message of at least 100 words and compare the result with other cryptography tools (like Cryptool 2) to verify.
2. Using the Playfair matrix below (Table 1.2) to encrypt the following message.

Message: ***I only regret that I have but one life to give for my country.***

This message is by Nathan Hale, a soldier in the American Revolutionary War.

Hàm generate_key_matrix

```
def generate_key_matrix(key):
    key = key.upper().replace("J", "I")
    key_matrix = [['' for _ in range(5)] for _ in range(5)]
    key_set = set()

    i, j = 0, 0
    for letter in key + 'ABCDEFGHIKLMNOPQRSTUVWXYZ':
        if letter not in key_set:
            key_matrix[i][j] = letter
            key_set.add(letter)
            j += 1
            if j == 5:
                i += 1
                j = 0

    return key_matrix
```

- Hàm này tạo ra ma trận chìa khóa cho Playfair Cipher dựa trên khóa nhập vào.
- Khóa được chuyển thành chữ in hoa và ký tự 'J' được thay thế bằng 'I'.

- Chữ cái trong khóa được sắp xếp vào ma trận 5x5, mỗi chữ cái chỉ xuất hiện một lần.
- Ma trận kết quả được trả về.

Hàm find_position: Hàm này tìm vị trí (dòng, cột) của một ký tự trong ma trận Playfair.

```
def find_position(matrix, char):
    for i in range(5):
        for j in range(5):
            if matrix[i][j] == char:
                return i, j
```

Hàm playfair_encrypt

```
def playfair_encrypt(plain_text, key):
    key_matrix = generate_key_matrix(key)
    encrypted_text = ''

    # Preprocess plaintext
    plain_text = plain_text.upper().replace("J", "I")
    plain_text = [char for char in plain_text if char.isalpha()]

    # Add a placeholder letter between consecutive identical letters
    for i in range(1, len(plain_text), 2):
        if plain_text[i] == plain_text[i - 1]:
            plain_text.insert(i, 'X')

    if len(plain_text) % 2 != 0:
        plain_text.append('X')

    # Encrypt pairs of letters
    for i in range(0, len(plain_text), 2):
        char1, char2 = plain_text[i], plain_text[i + 1]
        row1, col1 = find_position(key_matrix, char1)
        row2, col2 = find_position(key_matrix, char2)

        if row1 == row2:
            encrypted_text += key_matrix[row1][(col1 + 1) % 5] + key_matrix[row2][(col2 + 1) % 5]
        elif col1 == col2:
            encrypted_text += key_matrix[(row1 + 1) % 5][col1] + key_matrix[(row2 + 1) % 5][col2]
        else:
            encrypted_text += key_matrix[row1][col2] + key_matrix[row2][col1]

    return encrypted_text
```

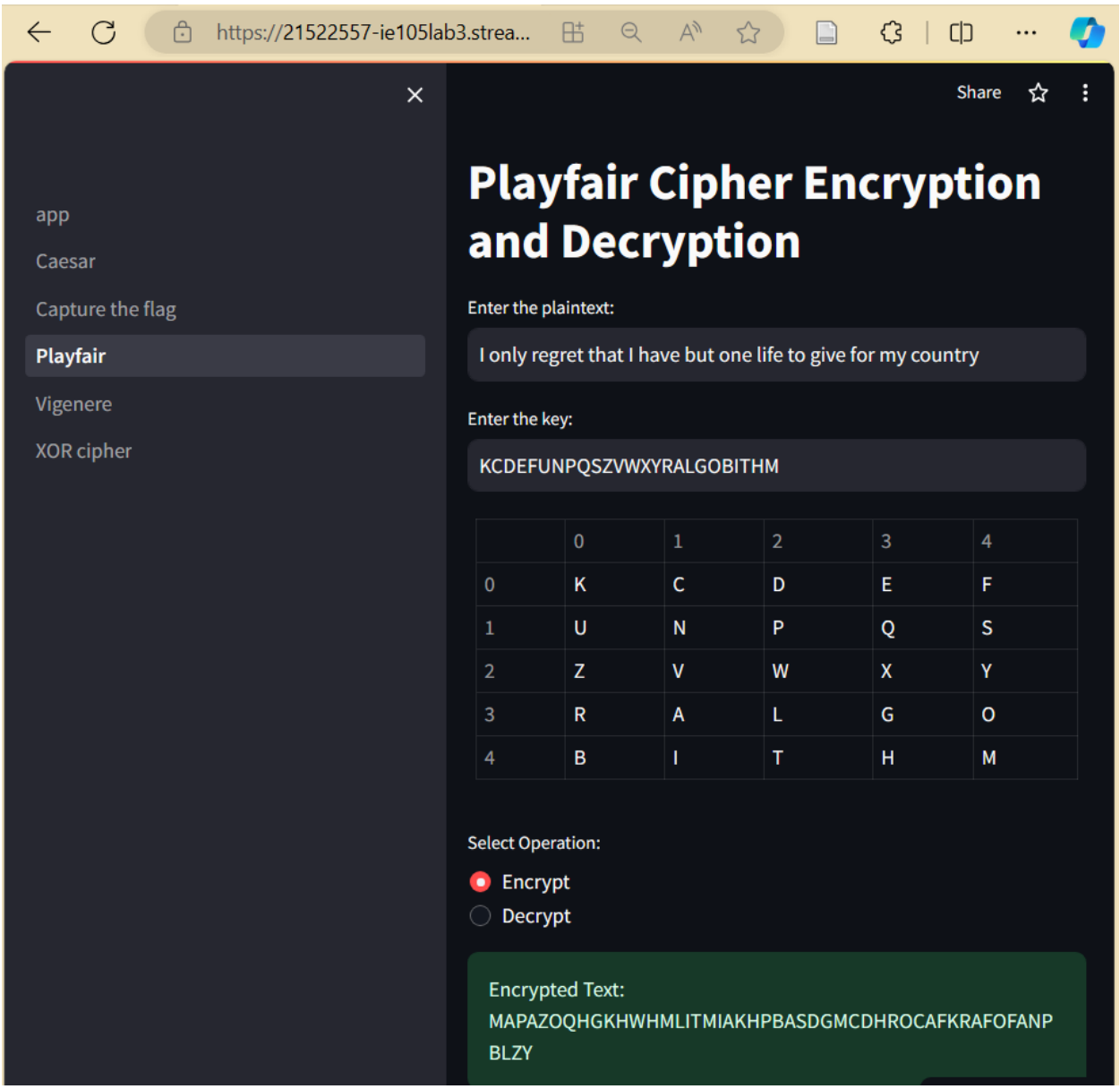
- Hàm này thực hiện quá trình mã hóa văn bản bằng Playfair Cipher.

- Văn bản đầu vào được tiền xử lý: chuyển đổi thành chữ in hoa, thay thế 'J' bằng 'I', và chỉ giữ lại các ký tự chữ cái.
- Một ký tự đặt chỗ ('X') được thêm giữa các cặp ký tự giống nhau.
- Mỗi cặp ký tự được mã hóa bằng cách xác định vị trí của chúng trong ma trận và áp dụng luật mã hóa của Playfair.

Hàm playfair_decrypt

```
def playfair_decrypt(encrypted_text, key):
    key_matrix = generate_key_matrix(key)
    decrypted_text = ''

    # Preprocess plaintext
    plain_text = encrypted_text.upper()
    plain_text = [char for char in plain_text if char.isalpha()]

    # Add a placeholder character if the length is odd
    if len(plain_text) % 2 == 1:
        plain_text.append('X')

    # Decrypt pairs of letters
    for i in range(0, len(plain_text), 2):
        char1, char2 = plain_text[i], plain_text[i + 1]
        row1, col1 = find_position(key_matrix, char1)
        row2, col2 = find_position(key_matrix, char2)

        if row1 == row2:
            decrypted_text += key_matrix[row1][(col1 - 1) % 5] + key_matrix[row2][(col2 - 1) % 5]
        elif col1 == col2:
            decrypted_text += key_matrix[(row1 - 1) % 5][col1] + key_matrix[(row2 - 1) % 5][col2]
        else:
            decrypted_text += key_matrix[row1][col2] + key_matrix[row2][col1]

    return decrypted_text
```

- Hàm này thực hiện quá trình giải mã văn bản bằng Playfair Cipher.
- Văn bản đã được mã hóa được tiền xử lý tương tự như trong hàm playfair_encrypt.
- Mỗi cặp ký tự được giải mã bằng cách xác định vị trí của chúng trong ma trận và áp dụng luật giải mã của Playfair.



Task 5: Write an application using your chosen programming language to encrypt and decrypt a message using Vigenère cipher.

Test your application by a message with at least 100 words and a key of about 10-20 letters. Then verify the result with other cryptography tools (e.g., Cryptool 2, dCode, etc.)

Hàm generate_vigenere_table

```
def generate_vigenere_table():
    alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    vigenere_table = np.zeros((26, 26), dtype='str')

    for i in range(26):
        for j in range(26):
            vigenere_table[i, j] = alphabet[(i + j) % 26]

    return vigenere_table
```

- Hàm này tạo ra bảng Vigenère, một ma trận 26x26 trong đó mỗi phần tử là kết quả của việc cộng chỉ số của hai chữ cái trong bảng chữ cái tiêu chuẩn.
- Bảng này sẽ được sử dụng trong quá trình mã hóa và giải mã.

Hàm vigenere_encrypt

```
def vigenere_encrypt(plaintext, key):
    alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    vigenere_table = generate_vigenere_table()
    ciphertext = ""

    key = key.upper()
    key_index = 0

    for char in plaintext:
        if char.isalpha():
            row = alphabet.index(key[key_index])
            col = alphabet.index(char.upper())

            ciphertext += vigenere_table[row, col]

            key_index = (key_index + 1) % len(key)
        else:
            ciphertext += char

    return ciphertext
```

- Hàm vigenere_encrypt thực hiện quá trình mã hóa Vigenère cho văn bản đầu vào.

- Dùng bảng Vigenère đã tạo và khóa, hàm lấy mỗi ký tự của văn bản đầu vào và thực hiện việc mã hóa bằng cách chọn giá trị tương ứng từ bảng Vigenère.
- Ký tự không phải là chữ cái sẽ được giữ nguyên.

Hàm vigenere_decrypt

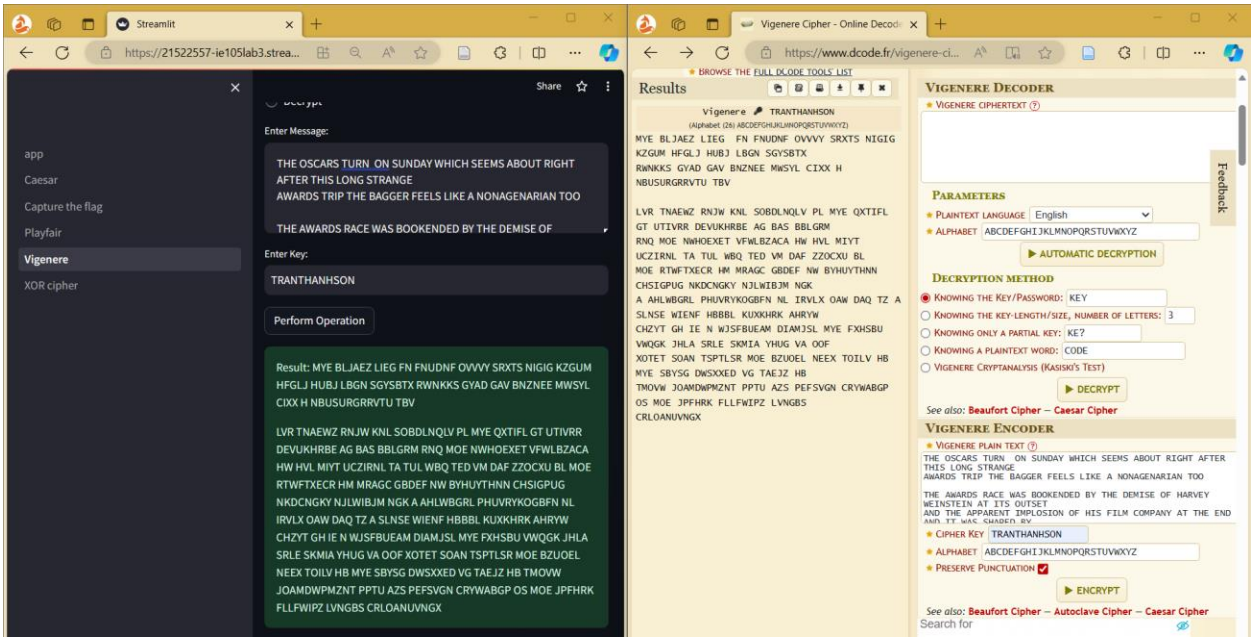
```
def vigenere_decrypt(ciphertext, key):
    alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    vigenere_table = generate_vigenere_table()
    plaintext = ""

    key = key.upper()
    key_index = 0

    for char in ciphertext:
        if char.isalpha():
            row = alphabet.index(key[key_index])
            col = np.where(vigenere_table[row, :] == char.upper())[0][0]

            plaintext += alphabet[col]

            key_index = (key_index + 1) % len(key)
        else:
            plaintext += char

    return plaintext
```

- Hàm vigenere_decrypt thực hiện quá trình giải mã Vigenère cho văn bản đã được mã hóa.
- Dùng bảng Vigenère và khóa, hàm lấy mỗi ký tự của văn bản đã mã hóa và thực hiện việc giải mã bằng cách xác định giá trị tương ứng từ bảng Vigenère.
- Ký tự không phải là chữ cái sẽ được giữ nguyên.

Ta thử so sánh kết quả bằng cách encrypt đoạn văn bên dưới bằng key “TRANTHANHSON”:

THE OSCARS TURN ON SUNDAY WHICH SEEMS ABOUT RIGHT AFTER THIS LONG STRANGE AWARDS TRIP THE BAGGER FEELS LIKE A NONAGENARIAN TOO

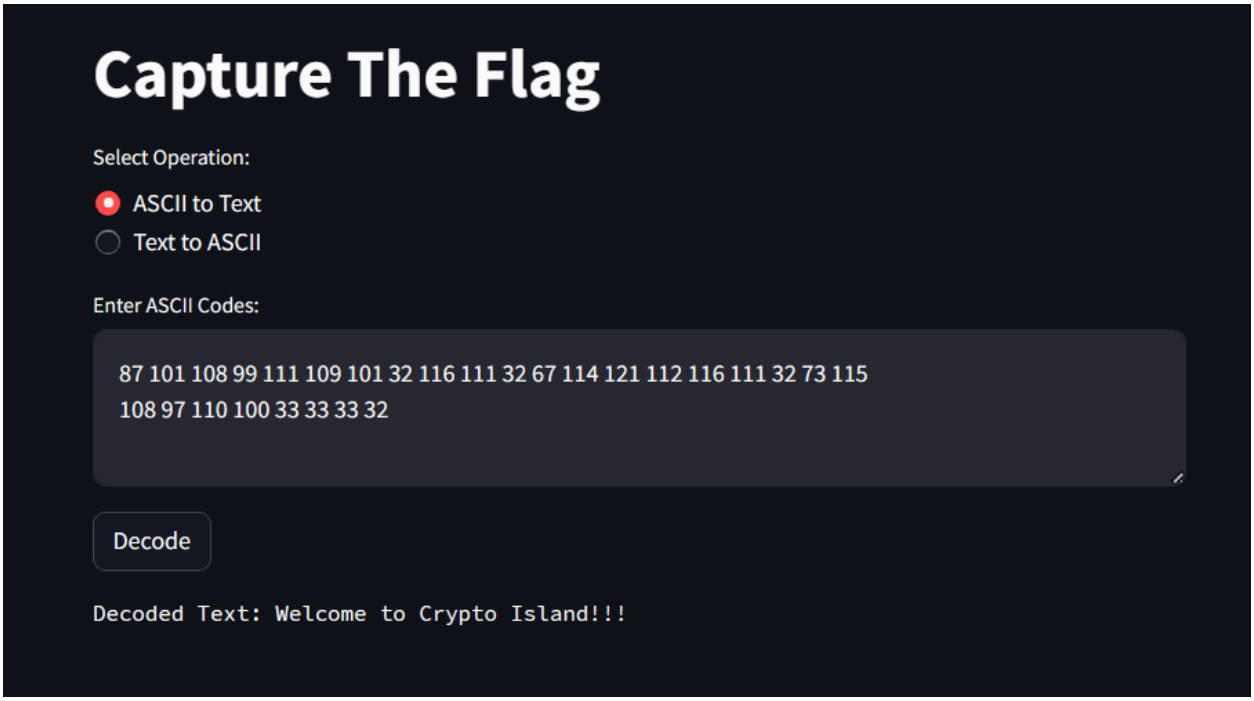
THE AWARDS RACE WAS BOOKENDED BY THE DEMISE OF HARVEY WEINSTEIN AT ITS OUTSET AND THE APPARENT IMPLOSION OF HIS FILM COMPANY AT THE END AND IT WAS SHAPED BY THE EMERGENCE OF METOO TIMES UP BLACKGOWN POLITICS ARMCANDY ACTIVISM AND A NATIONAL CONVERSATION AS BRIEF AND MAD AS A FEVER DREAM ABOUT WHETHER THERE OUGHT TO BE A PRESIDENT WINFREY THE SEASON DIDNT JUST SEEM EXTRA LONG IT WAS EXTRA LONG BECAUSE THE OSCARS WERE MOVED TO THE FIRST WEEKEND IN MARCH TO AVOID CONFLICTING WITH THE CLOSING CEREMONY OF THE WINTER OLYMPICS THANKS PYEONGCHANG



Task 6: Decode the following text to find the flag. It is known that the message was encoded with ASCII code:

87 101 108 99 111 109 101 32 116 111 32 67 114 121 112 116 111 32 73 115

108 97 110 100 33 33 33 32



Task 7: The file crypto01.jpg can be downloaded from **here**, and it contains a flag. Knowing that this image was encrypted with XOR cipher by a 6-letter key. Let's find the flag and describe how to do

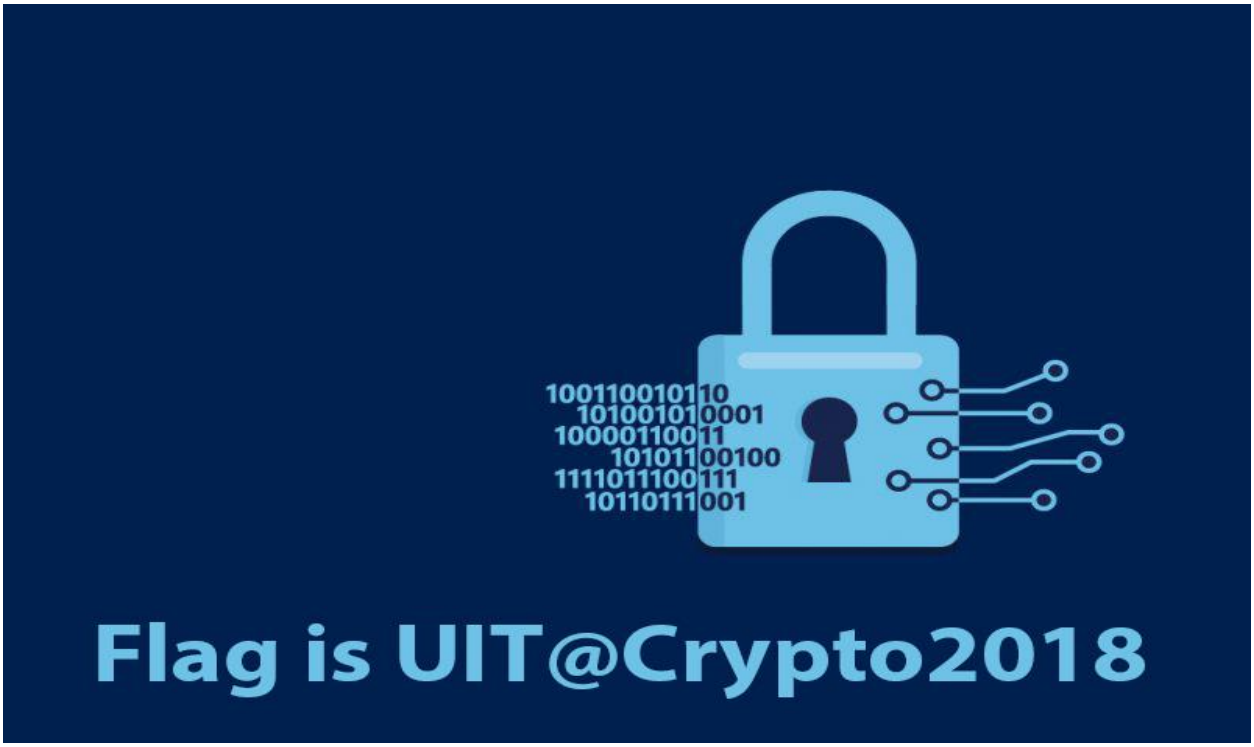
- Mở tệp "crypto01.jpg" bằng Notepad và thấy chuỗi "hcmuit", "uithcm", "HCMUIT", "UITHCM",...
- Dựa vào đó, ta đã dự đoán khóa có thể là "hcmuit" hoặc "uithcm".

```
def decrypt(ciphertext, key):
    # Hàm giải mã với phép XOR
    decrypted = bytearray()  # Khởi tạo một mảng byte để lưu dữ liệu đã giải mã
    key_len = len(key)  # Độ dài của khóa
    for i, byte in enumerate(ciphertext):
        # Thực hiện phép XOR giữa byte trong ciphertext và byte tương ứng trong khóa
        decrypted_byte = byte ^ key[i % key_len]
        decrypted.append(decrypted_byte)  # Thêm byte đã giải mã vào mảng
```

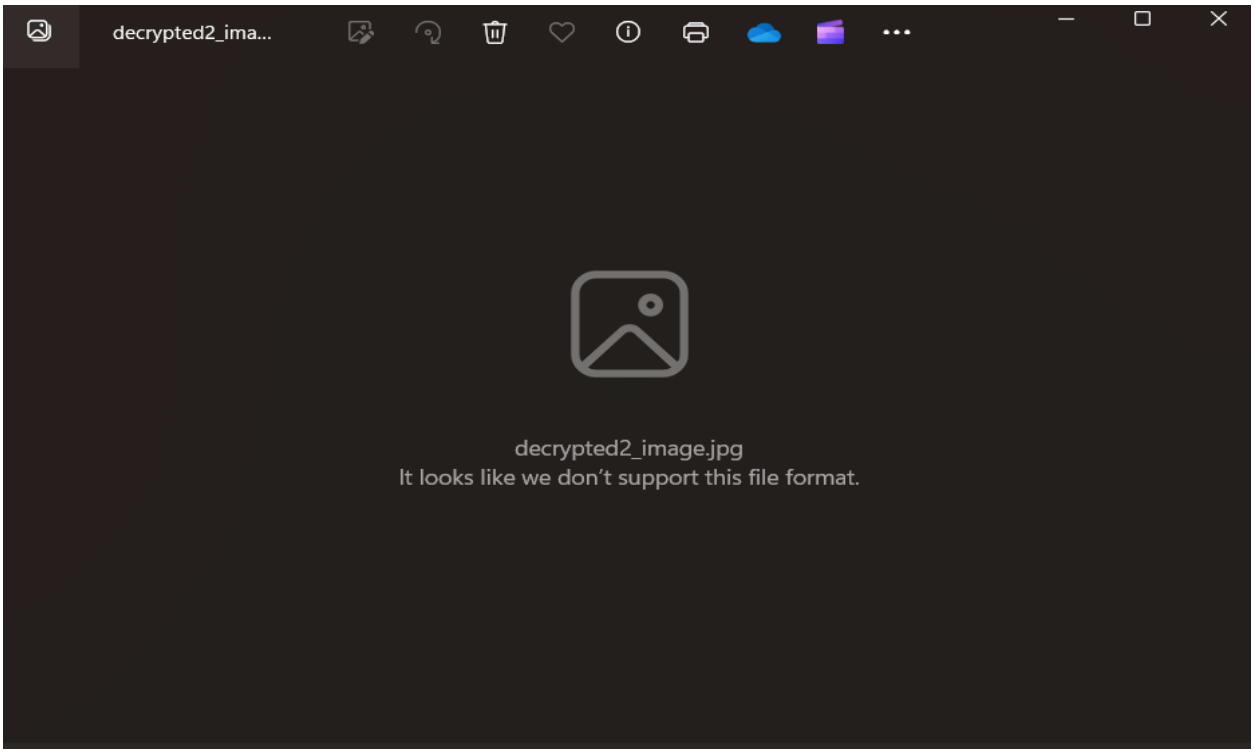
```
    return bytes(decrypted)  # Trả về dữ liệu đã giải mã dưới dạng bytes

# Đọc dữ liệu từ tệp hình ảnh đã mã hóa
with open("crypto01.jpg", "rb") as file:
    encrypted_data = file.read()

# Khai báo hai khóa dự đoán
encryption_key1 = b'uithcm'
encryption_key2 = b'hcmuit'

# Giải mã với hai khóa dự đoán
decrypted1 = decrypt(encrypted_data, encryption_key1)
decrypted2 = decrypt(encrypted_data, encryption_key2)

# Lưu kết quả giải mã vào hai tệp hình ảnh đã giải mã tương ứng
with open("decrypted1_image.jpg", "wb") as file1:
    file1.write(decrypted1)
with open("decrypted2_image.jpg", "wb") as file2:
    file2.write(decrypted2)

# In ra thông báo khi quá trình giải mã hoàn tất
print("Finish")
```

Output:

decrypted1_image.jpg with key = “uithcm”



decrypted2_image.jpg with key = “hcmuit”



⇨ Key =’uithcm’