Q1 Team name
0 Points

Kerberos

Q2 Commands
10 Points

List the commands used in the game to reach the ciphertext.

'enter' 'enter' 'pick' 'c' 'c' 'back' 'give'
'back' 'back' 'thrnxxtzy' 'read'

Q3 Analysis
50 Points

Give a detailed analysis of how you figured out the password? (Explain in less than 500 words)

Using % as modulus operator
using p = prime number = 198070406285660843983859875 81

We solved the question mainly using the modular arithmetic and one brute force method to calculate the value of g and then password.

---> Given the pairs (a, b) of the form b = password * g ^ a .
112268153502635318149633363 15 = password * g ^ 324 -eqn 1
919054866790027430083039122 0 = password * g ^ 2345 -eqn 2
413865262965561357081900049 7 = password * g ^ 9513 -eqn 3

---> Dividing one equation with other
(g^A1 / g^A2) = (B1 / B2) %p
=> g^(A1-A2) = (B1 * B2^-1) %p
=> g^(A12) = (B21) %p
So the equation we got were
g^2021 = 702128436930163864057706667 9 %p -eqn 21) Dividing 2 by 1
g^7168 = 633924885173732750892405925 7 %p -eqn 32) Dividing 3 by 2
g^9189 = 342634738514499522582501678 1 %p -eqn 31) Dividing 3 by 1

---> "BRUTE FORCE/ HIT AND TRIAL FUNCTION" --
Assuming that if by dividing or multiplying in some order the above three expression, if the exponent on g becomes 1 we can get value of g directly. So i tried to find possible set of values(i, j, k) such that +- a31 * i +- a32 * j +- a21 * k == 1 using a hardcoded program.
And we found many sets of such values.

We picked one set (139,0,632) as it had a 0 for a32 which made calculation easier
-9189*139 + 632*2021 = 1

---> Therefore after appropriate multiplication and division, we got 2 new expressions
From 31
g ^ (a31 * 139) = (g ^ a31)^ 139 = b31 ^ 139 %p
=> g ^ 1277271 = 342634738514499522582501678 1^139 %p = 17064457453994872811494067145

From 21
g ^ (a21 * 632) = (g ^ a21)^ 632 = b21 ^ 632 %p
=> g ^ 1277272 = 702128436930163864057706667 9^632%p = 91457147351611408993901999 31

Dividing above two equations
g^ (1277272 - 1277271) = (91457147351611408993901999 31 * (17064457453994872811494067145 ^-1)%p)%p
=> g = 192847283928500239481729

*** Here we matched this g with the given g = 1___4_2_____0___94____9 with missing entries and it matched.

---> Finally we computed password using the eqn 1 by substituting value of g
password = (b1 * (g^a1)^-1) % p
password = 360852885036840078603 6725

--NOTE: I may have missed %p in some places. Please forgive for missing %p in those places for the mathematical equations.

Q4 Password

10 Points

What was the final command used to clear this level?

360852885036840078603 6725

Q5 Codes

0 Points

Upload any code that you have used to solve this level.

▾ Assignment_3.ipynb Download

In [37]:

```
#Input Given Pair of (a, pass*g^a) as (a,b)
# P is the prime number belonging to the field Zp*
```

```
p = 19807040628566084398385987581
(a1,b1) = (324,  1122681535026353181496333631 5)  #--1
(a2,b2) = (2345, 9190548667900274300830391220)   #--2
(a3,b3) = (9513, 4138652629655613570819000497)   #--3
```

In [4]:

```
# Simplifier :  Dividing one equation with other
# g^a1 / g^a2 (mod p) = b1 / b2 (mod p)
# => g^(a1-a2) (mod p) = (b1 * b2^-1) mod p

def eqndivider( a1,a2,b1,b2 ):
    a = a1-a2
    b = ( b1 * pow(b2,-1,p) ) %p
    return(a,b)
```

In [38]:

```
# Computing equation for all possible pair from given 3 sets of
points
# aij means ai/aj

a21,b21 = eqndivider(a2,a1,b2,b1)
a32,b32 = eqndivider(a3,a2,b3,b2)
a31,b31 = eqndivider(a3,a1,b3,b1)
```

In [40]:

```
# PRINT
a21,a32,a31,b21,b32,b31
```

Out [40]:

```
(2021,
 7168,
 9189,
 702128436930163864057706 6679,
 633924885173732750892405 9257,
 342634738514499522582501 6781)
```

In [ ]:

In [45]:

```
# ------ Hit and Trial -----
# My aim was to somehow make the exponent in g^e  e = 1.
# So I tried to search for set of values ( i, j, k) such that
# any combination of the +- e1 * i +- e2 * j +- e3 * k  becomes 1.
# Luckily i found many.  Below is the code
# -
# Also stopped the execution after getting sufficient information

for i in range(1000):
    for j in range(1000):
        for k in range(1000):
             if( -a31 * i + a32 * j + a21 * k == 1 ):
                    print(i,'   ',j,'    ',k)
                    break
```

```
139    0     632
140    1     633
141    2     634
142    3     635
143    4     636
144    5     637
145    6     638
146    7     639
147    8     640
148    9     641
149    10     642
150    11     643
151    12     644
152    13     645
153    14     646
```

```
154    15     647
155    16     648
156    17     649


---------------------------------------------------------------------------
------KeyboardInterrupt                         Traceback (most
recent call last)<ipython-input-45-9269dfc3af53> in <module>
      4     for j in range(1000):
      5         for k in range(1000):
----> 6             if( -a31 * i + a32 * j + a21 * k == 1 ):
      7                 print(i,'  ',j,'   ',k)
      8                 break
KeyboardInterrupt:
```

In [60]:

```
# Since a21 * 632 - 139 * a31  = 1
# Adding 632 and 139 as exponents in both side for each eqaution
makes
# g ^ ( a31 * 139 ) = (g ^ a31)^ 139 = b31 ^ 139 = num31
# g ^ ( a21 * 632 ) = (g ^ a21)^ 632 = b21 ^ 632 = num21

num31 = pow(b31, 139, p)
num21 = pow(b21, 632, p)

print('num31 = ',num31,'\nnum21 =',num21)
```

```
num31 =  170644574539948728114940671 45
num21 = 914571473516114089939019 9931
```

In [61]:

```
# Finally divding two updated equation
# num31 / num 21 = g ^ ( a21* 632 - a31*139 ) = g

g = ( num21 * pow(num31,-1,p) ) %p
print('g = ', g)
```

```
g =  192847283928500239481729
```

In [62]:

```
# Verifier that g matches the partial characters given in question

# Calculating password using equation 1

password = (b1 * pow( pow(g,a1,p) , -1, p ) ) %p
print('Password = ',password)
```

```
Password =  360852885036840078603 6725
```

In [ ]:

Assignment 3

GRADED

GROUP

SUMIT SINGH SHEORAN
MANI KANT KUMAR
AMAN ARYAN
View or edit group

TOTAL POINTS

70 / 70 pts

QUESTION 1

Team name **0** / 0 pts

QUESTION 2

Commands **10** / 10 pts

QUESTION 3

Analysis **50** / 50 pts

QUESTION 4

Password **10** / 10 pts

QUESTION 5

Codes **0** / 0 pts