Q1 Team Name
0 Points

INFINITY

Q2 Commands
10 Points

List the commands used in the game to reach the ciphertext

go, enter, pluck, 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)

Let's say:
$(a, x) = (429, 431955503618234519808008749742)$
$(b, y) = (1973, 176325509039323911968355873643)$
$(c, z) = (7596, 98486971404861992487294722613)$
Also $p = 455470209427676832372575348833$
Also, say password is represented by $\mathbf{P}$.
All the calculations are being done in the prime field of prime p.
Given:
$Pg^a = x \qquad [1]$
$Pg^b = y \qquad [2]$
$Pg^c = z \qquad [3]$
As $gcd(x, p), gcd(y, p)\ and\ gcd(z, p)$ is 1, therefore their inverse exists.
Dividing equation $\mathbf{2}$ by equation $\mathbf{1}$ and equation $\mathbf{3}$ by equation $\mathbf{2}$, we get:
$g^{b-a} = yx^{-1}$
$g^{c-b} = zy^{-1}$
Let $m = b - a = \mathbf{1544}$ and $n = c - b = \mathbf{5623}$, where m and n are the two distinct powers of the g, then above equations become:
$g^m = yx^{-1} \qquad [4]$
$g^n = zy^{-1} \qquad [5]$
As $GCD(m, n)$ is 1, then according to $Extended\ Euclidean\ Algorithm$ there exists i and j such that:
$mi + nj = gcd(m, n) = 1$
Thus we get $\mathbf{i=-2298}$ and $\mathbf{j=631}$
Taking i powers of equation $\mathbf{4}$ and j powers of equation $\mathbf{5}$ and multiplying them together, we get:
$(g^m)^i (g^n)^j = (yx^{-1})^i (zy^{-1})^j$
$g^{mi+nj} = x^{-i} y^{i-j} z^j$
Therefore, $g = x^{-i} y^{i-j} z^j$
On putting the values of i, j, x, y, z we get $\mathbf{g=525650854179633110276943339}$
The GCD of g and p also comes to be 1, hence its inverse also exists.
We get $g^{-1} = \mathbf{386506431948388234043577899453}$
Hence, password can be obtained as:
$P = xg^{-a} = yg^{-b} = zg^{-c}$
On putting values of g^{-1}, x, a we get, $\mathbf{P=134721542097659029845273957}$

Q4 Password
10 Points

What was the final command used to clear this level?

134721542097659029845273957

Q5 Codes
0 Points

Upload any code that you have used to solve this level

INFINITY_CS641a.py Download

```
def euclidean_gcd(m, n):
    if m == 0:
        return (0, 1)
    else:
        x, y = euclidean_gcd(n % m, m)
        return (y - (n//m) * x, x)



(a,x) = (429,431955503618234519808008749742)
(b,y) = (1973,176325509039323911968355873643)
(c,z) = (7596,98486971404861992487294722613)
p = 455470209427676832372575348833

# Powers of g = (m=a-b and n=c-b)
m = b-a
n = c-b
print("m = ", m ,"and n = ", n)

# Using Extended Euclidean theorem, we got i and j such that mi+nj=1
i, j = euclidean_gcd(m, n)
print("i = ", i ,"and j = ", j)

x_inverse, _ = euclidean_gcd(x, p)
y_inverse, _ = euclidean_gcd(y, p)
z_inverse, _ = euclidean_gcd(z, p)
print("x_inverse = ", x_inverse ,", y_inverse = ", y_inverse, "and z_inverse = ", z_inverse)

#i is positive and j is negative
g = (pow(x_inverse, i, p)*pow(y, i-j, p)*pow(z_inverse, -1*j, p))%p
print("g = ", g)

g_inverse, _ = euclidean_gcd(g, p)
print("g_inverse = ", g_inverse)

password = (x * pow(g_inverse, a, p))%p
print("password = ", password)


# Output
# m =  1544 and n =  5623
# i =  -2298 and j =  631
# x_inverse =  7074996790223471732904681640 , y_inverse =  -226523059948924229766221663708 and z_inverse =  105171748371597409614445194812
# g =  525650854179633110276943339
# g_inverse =  -68963777479288598328997449380
# password =  134721542097659029845273957


```