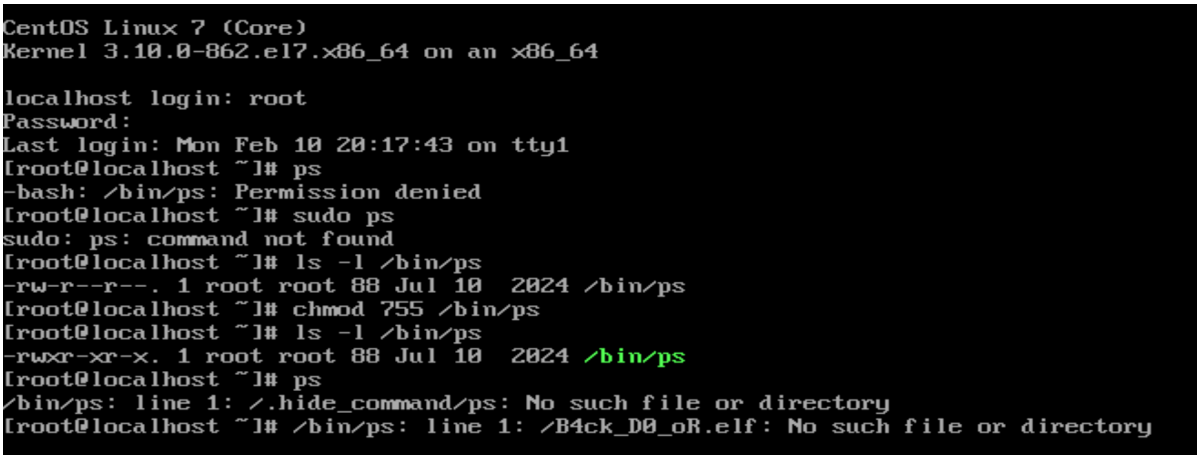
HGAME2025-Week2-WP

队伍名：gcssjcsa，队伍ID：0x000011

Misc

Computer cleaner plus



flag `hgame{B4ck_DO_oR}`

Invest in hints

猜测0和1代表该hint是否有flag对应位的字符，1为有、0为无，按以下代码找出拼凑flag所需数量较少的hint组

```
hints =
["000011001010011111010000000010010001110100000000000000000001101111000100",

"011010001110110000000000101000100001001101100000000010010001110011000000",

"101001000000010110001100010011010000100011010111010101100010000000000000",

"000010100000100100001001100001000000100001001011001110000010111000001111",

"011100101001001000000000000000000011010110011000001111000101100000001000",

"011101000010010000100101111011110111010010001000100110010000100111000000",

"100001010100000000110000011001010010101101000001101100100010001000110000",

"000001111010000010010000011001001000001100001100001010000011011101000000",

"010011010010010000000010010011101000000000000010110001000100001010101010",

"100100101001100110111000100110011001001000011100100101010010001000001111",

"010010001000110000010000000000110100011100010000001011000010001000010100",

"001010000100001110001011100000100010000000010001111000100011010010011101",
```

```
"0100001011101000000001010000101000101100010010001000000000000000001000000",

"0111011011001100000001000001100000001000000000000011100000001000001000 1",

"0110000000001100011000000001000100000000001100110000011001000101101000 0",

"0111001100100010100110000101100001101000000110001010000001101000000 1000",

"0011101100001100000010010010100010010010100001000110011100100010000 1000",

"0100011001010101110011010111001000111110001101000000010101010000000 10010",

"1111101011100011010001000000001000110111101001101000110000001100000 1001",

"0000001011010110010010001100101101100110000010001001111100001100000 1101",

"0000110000111010100001011100110001110010001001110000101000000000 1000010",

"0110000000001100100101110000010100011011100010110001010111100000 1010100",

"0000100000101001000000110101011011000011011101101110010101111001 0110000",

"0101001010000000011101111000100001011010000100011100110101010000000 10000",

"1101000001100001010000101000011101101010000111101010010010000011111 0110"]

valid_char_num = []
grouped_hints_idxs = [[] for _ in range(71)]
flag_bit =
0b0000000000000000000000000000000000000000000000000000000000000000000
choosed_hints_idxs = []

def is_finished():
    return flag_bit ==
0b1111111111111111111111111111111111111111111111111111111111111111111

# 计算每个hint的有效位数
for hint in hints:
    valid_char_num.append(hint.count("1"))

# 按flag的每一位分组hint
for i in range(71):
    for j in range(25):
        if hints[j][i] == "1":
            grouped_hints_idxs[i].append(j)

while not is_finished():
    min_group = min(grouped_hints_idxs, key=len) # 取出flag一位中有效hint数最少的组
    max_hint_idx = max(min_group, key=lambda hint: valid_char_num[hint]) # 取出该
组中有效位数最多的hint
    choosed_hints_idxs.append(max_hint_idx)
    flag_bit |= int(hints[max_hint_idx], 2)
    groups_to_remove = []
    for group in grouped_hints_idxs:
        if max_hint_idx in group:
```

```
            for idx in group:
                if idx != max_hint_idx:
                    valid_char_num[idx] -= 1 # 和选中hint有共同有效位的位置不再有效
            groups_to_remove.append(group)
    for group in groups_to_remove:
        grouped_hints_idxs.remove(group) # 移除已选hint的有效位的组

print(choosed_hints_idxs) # [24, 16, 17, 6, 18]
```

输出的是索引值，实际hint要+1
由于在写wp时重写了代码，解题用的最后一个hint和wp的代码输出的不是同一个，实际上是等效的
解锁对应hint，观察发现二进制数和对应字符是相反的，由以下代码合成flag

```
choosed_hints = {24: "gae{Aug39iLOgQSyAq3i69lbRmLEa2}",
         16: "mgko99i7gayzt1hCuADLdHa4",
         17: "geMfo9gSCy2Azq6DhNCxuR2mA5Edm2",
         6: "megk9CiLrKWyAqi9hN8rELm}",
         19: "ham5Yo99ACLQy2q3i61NluRCA5Ewd"}

for i in range(71):
    for j in choosed_hints.keys():
        if hints[j][len(hints[j]) - i - 1] == "1":
            index = hints[j].count("1", len(hints[j]) - i)
            print(choosed_hints[j][index], end="")
            break

# hgame{Aug5YMkf3o99ACi7LrOgQSCKaWy2Azq3ti691DhNlCbxu8rR2mCAD5LEwLdmHa42}
```

flag `hgame{Aug5YMkf3o99ACi7LrOgQSCKaWy2Azq3ti691DhNlCbxu8rR2mCAD5LEwLdmHa42}`

Web

Level 21096 HoneyPot

Password字段是没有验证的，导入数据时remote_password设置为123;/writeflag即可执行/writeflag命令，访问/flag即得**flag** `hgame{c06f4286-38c3-4d66-4784-135029d53931}`

Crypto

Ancient Recall

```
# 定义牌组（注意空格在 flag 中会替换为下划线）
Major_Arcana = [
    "The Fool", "The Magician", "The High Priestess", "The Empress",
    "The Emperor", "The Hierophant", "The Lovers", "The Chariot",
    "Strength", "The Hermit", "Wheel of Fortune", "Justice",
    "The Hanged Man", "Death", "Temperance", "The Devil",
    "The Tower", "The Star", "The Moon", "The Sun",
    "Judgement", "The World"
]

wands = [
    "Ace of Wands", "Two of Wands", "Three of Wands", "Four of Wands",
```

```
    "Five of Wands", "Six of Wands", "Seven of Wands", "Eight of Wands",
    "Nine of Wands", "Ten of Wands", "Page of Wands", "Knight of Wands",
    "Queen of Wands", "King of Wands"
]
cups = [
    "Ace of Cups", "Two of Cups", "Three of Cups", "Four of Cups",
    "Five of Cups", "Six of Cups", "Seven of Cups", "Eight of Cups",
    "Nine of Cups", "Ten of Cups", "Page of Cups", "Knight of Cups",
    "Queen of Cups", "King of Cups"
]
swords = [
    "Ace of Swords", "Two of Swords", "Three of Swords", "Four of Swords",
    "Five of Swords", "Six of Swords", "Seven of Swords", "Eight of Swords",
    "Nine of Swords", "Ten of Swords", "Page of Swords", "Knight of Swords",
    "Queen of Swords", "King of Swords"
]
pentacles = [
    "Ace of Pentacles", "Two of Pentacles", "Three of Pentacles", "Four of
Pentacles",
    "Five of Pentacles", "Six of Pentacles", "Seven of Pentacles", "Eight of
Pentacles",
    "Nine of Pentacles", "Ten of Pentacles", "Page of Pentacles", "Knight of
Pentacles",
    "Queen of Pentacles", "King of Pentacles"
]

Minor_Arcana = wands + cups + swords + pentacles
tarot = Major_Arcana + Minor_Arcana

# 定义一次轮盘运算的逆运算
def inverse_fortune(w):
    # w 为长度为 5 的列表
    total = sum(w)
    S = total // 2  # S = v0+...+v4
    v0 = (w[0] + w[2] + w[4] - w[1] - w[3]) // 2
    v1 = w[0] - v0
    v2 = w[1] - v1
    v3 = w[2] - v2
    v4 = w[3] - v3
    return [v0, v1, v2, v3, v4]

# 给定最终运算后的 5 个大数（题目中提供的输出）
F = [
    25329519520662917748904983691141959172407947049182105205710670853114746750 19,
    25329519520662917748903276660741003578980230131054431788812947003815097952 70,
    25329519520662917748905544592872766049031303158592585441730683769670723357 30,
    25329519520662917748908653282415328853915101626115345140144091742842991390 15,
    25329519520662917748908306626081341560179463763099899341758339139211426093 34
]

# 对 F 反复应用逆运算 250 次，得到最初的向量 v
v = F
for _ in range(250):
    v = inverse_fortune(v)

# 根据 v 恢复初始牌（注意大阿卡那逆位用 "re-" 前缀）
```

```
initial_fate = []
for x in v:
    if x < 0:
        card_index = -x - 1  # 因为逆位时 v = -index - 1
        card = "re-" + tarot[card_index]
    else:
        card = tarot[x]
    initial_fate.append(card)

print(initial_fate)
# 构造 flag
flag = "hgame{" + "&".join(initial_fate) + "}"
flag = flag.replace(" ", "_")
print("Flag is:")
print(flag)
```

flag `hgame{re-The_Moon&re-The_Sun&Judgement&re-Temperance&Six_of_Cups}`