Jesus Blanco
Ankita Koratkar

CPE 123-29, Lab 2

File: hidden_in_plain_sight.txt

```
f=open(“hidden_in_plain_sight.txt”)
#Opens and reads the file hidden_in_plain_sight
filelines = f.readlines()
found_flag=False
closed_flag=False
flag = "flag"
for line in filelines:
    for i in line:
        #If the end of the flag is found
        if(i=="}"):
                closed_flag = True
                flag=flag+i
                break
        #If the beginning of the flag is found
        if(i=="{"):
                found_flag=True
                flag=flag+i
        #If the current character is within the two brackets the character is added to the flag
        elif(found_flag):
                flag = flag + i
        else:
                pass
    #If the portion containing the flag is over, break out of the loop
    if(closed_flag):
        break
print flag
f.close()
quit()
```

Answer : flag{you_found_me}

File: one_plus_one_equals_flag.txt

```
f=open("one_plus_one_equals_flag.txt")
#Opens and reads the file one_plus_one_equals_flag.txt
filelines = f.readlines()
#the number of spaces between the characters needed
count = 0
#iteration the loop is on
p=0
#the index of the character that needs to be printed
currentcount = 0
#the flag that needs to be found
finalflag=""
for line in filelines:
        for a in line:
            #Saves the character if the iteration loop is on matches the number of the character
            if(p==currentcount):
                finalflag=finalflag+a
                #increments the number of spaces between the characters by one
                count = count +1
                #increases the number for the character that needs to be printed by count
                currentcount = currentcount+count
            p=p+1

print finalflag
f.close()
quit()
```

Answer: flag{easy_peasy_lemon_squeezy}

File: quad-radical.txt

```
f=open("quad-radical.txt")
filelines = f.readlines()
#the number that needs to be squared to get the number of spaces between the characters that
#need to be saved
square = 0
#iteration the loop is on
p=0
#the index the character that needs to be printed
currentcount = 0
finalflag=""
for line in filelines:
    for a in line:
       #Saves the character if the iteration loop matches the number of the character
        if(p==currentcount):
            finalflag=finalflag+a
            #if statement just for the first iteration to get the next character number
            if(p==0):
                currentcount = currentcount + 1
            else:
                #increments the number needed to be squared by one
                square = square +1
                #current count is changed to the number of the character that needs to be saved next
                currentcount = currentcount + (square*square) + 1
        p=p+1
print finalflag
f.close()
quit()
```

Answer: flag{this_is_a_flag_or_is_it}

File: I_fibbed.txt

```
f=open("I_fibbed.txt")
filelines = f.readlines()
#the first number of the fibonacci series
first = 0
#the second number of the fibonacci series
second = 1
#third will later be the sum of the past two numbers in the fibonacci series
third = 0
#iteration the loop is on
p=0
finalflag=""
#the index of the character that needs to be saved
finalcount=0
for line in filelines:
    for a in line:
        #Saves the character if the iteration loop matches the number of the character
        if(p==finalcount):
            finalflag = finalflag+a
            # if it’s the 0th iteration nothing changes,  just the 0th character is saved
            if(p==0):
                pass
            #if its the 1st iteration final count is incremented by one to save the 1st character
            elif(p==1):
                finalcount =finalcount+1
            else:
                #following the fibonacci sequence the third number is the sum of the last two
                third = first + second
                # first takes the value of the second number
                first = second
                #second takes the value of the third number
                second = third
              #finalcount is updated with the number of the next character
              finalcount = finalcount + third +1
        p=p+1

print finalflag
f.close()
quit()
```

Answer : flag{time_for_some_fibonachos}

File: bumps_on_a_log_2.txt

```
f=open("bumps_on_a_log_2.txt")
fileline = f.readline()
power = -1
#iteration the loop is on
p=0
#the index of the character that needs to be saved
currentcount = 0
finalflag=""
for a in fileline:
    if(p==currentcount):
        finalflag=finalflag+a
        #the power to which 2 is raised to is increased by one
        power = power +1
        #currentcount is updated with the number of the next character
        currentcount = currentcount + 2**power + 1
    p=p+1
print finalflag
f.close()
quit()
```

Answer : flag{that_escalated_quickly}