200021118_EEE_4710_Assignment_02

October 13, 2024

Islamic University of Technology (IUT)
Organization of Islamic Cooperation (OIC)
Department of Electrical and Electronic Engineering

ID	**200021118**
Name	**Omar Faruk**
Course ID	**EEE 4710**
Course Name	**Artificial Intelligence and Machine Learning Lab**
Assignment No.	**2**

[1]:
```
import numpy as np
import pandas as pd
```

1 Question - 01

A numpy array is given. Find the maximum value of -

- the whole matrix
- column-wise
- row-wise

[2]:
```
mat= np.random.randint(3,100,(3,7))
mat2 = np.random.randint(200,2000,(1,5))

# write your answer below
```

[3]:
```
# Solution
def q1_max(mat):
```

```
    print(f"mat=\n{mat}\n")
    print(f"a. max of whole matrix: {mat.max()}")
    print(f"b. column wise max: {mat.max(axis=0)}")
    print(f"c. row wise max: {mat.max(axis=1)}\n")

# For mat
q1_max(mat)

#  For mat2
q1_max(mat2)
```

```
mat=
[[81 87 35 96 13 48 80]
 [96 34 32 70 93 46 33]
 [63 35 95 69 78 32 21]]

a. max of whole matrix: 96
b. column wise max: [96 87 95 96 93 48 80]
c. row wise max: [96 96 95]

mat=
[[1200 1940 1594 1442  333]]

a. max of whole matrix: 1940
b. column wise max: [1200 1940 1594 1442  333]
c. row wise max: [1940]
```

2 Question - 02

Abbreviate the given strings.

[4]:
```
s1= 'dhaka University'  # output= 'DU'
s2= 'Laser Interferometer Gravitational-Wave Observatory'   # output = 'LIGO'

# write your answer below
```

[5]:
```
# Solution
def q2_abbreviate(s: str) -> str:
    s = [ x[0] for x in s.split(' ') ]
    s = ''.join(s).upper()
    print(s)
    # return s

# For s1
q2_abbreviate(s1)
```

```
# For s2
q2_abbreviate(s2)
```

```
DU
LIGO
```

3 Question - 03

Given a tuple of integers, sort them based on the absolute value of the numbers. The output should also be a tuple.

[6]:
```
m=(3,4,0,-4,-2,1,6)    # output = (0, 1, -2, 3, 4, -4, 6)

# write your answer below -
```

[7]:
```
# Solution
def q3_sort(m: tuple) -> tuple:
    # tuple to list
    m = list(m)
    # sort based on absolute value
    m.sort(key=abs)
    # list to tuple
    m = tuple(m)
    print(m)
    # return m

# For m
q3_sort(m)
```

```
(0, 1, -2, 3, 4, -4, 6)
```

4 Question - 04

A list of length 2n is given.

a=[x1,x2,….,xn,y1,y2,…,yn].

You need to create a dataframe object from the given list with two columns. Each column will have half the items of the list. Column-1 will contain x1, x2, …., xn. Column-2 will contain y1, y2, …., yn.

Name of the columns should be - Col1 and Col2.

[8]:
```
a=[3,6,1,2,'a','c','b','z']

# write your answer below -
```

[9]:
```
# Solution
def q4_dataframe(a: list) -> pd.DataFrame:
```

```
    m = int(len(a)/2)
    x = a[:m]
    y = a[m:]
    df = pd.DataFrame({
        'Cal1': x,
        'Cal2': y
    })
    return df

# For a
q4_dataframe(a)
```

[9]:

	Cal1	Cal2
0	3	a
1	6	c
2	1	b
3	2	z

5 Question - 05

A numpy array is given. Write a function that produces two outputs -

- the index of all non-zero elements in the array
- the index of all positive elements in the array

[10]:
```
# write your answer below -

def find_elem(a):
    ind_non_zero = np.nonzero(a!=0)
    ind_pos = np.nonzero(a >= 0)

    return ind_non_zero, ind_pos
```

[11]:
```
a= np.random.randint(-30,10,(9,7))
b = np.random.randint(-30,0,(5,12))

# call your function here
```

[12]:
```
z_a , p_a = find_elem(a)
print(f"a = \n{a}\n")
print(f"Non-zero indices: {z_a}\n")
print(f"Positive indices: {p_a}\n\n")

z_b , p_b = find_elem(b)
print(f"b = \n{b}\n")
print(f"Non-zero indices: {z_b}\n")
print(f"Positive indices: {p_b}\n\n")
```

```
a =
[[ -9 -12   0 -24   1 -11   1]
 [-27 -20 -21 -20 -19   5 -13]
 [ -8  -7   3   9  -6  -3  -1]
 [-23 -27   8   8 -29 -12   9]
 [  3   6 -30   7   8  -7 -20]
 [ -7 -12   5 -17 -29 -12 -21]
 [  9 -14   9 -14   9 -30 -15]
 [ -5 -18  -5 -27 -28  -7 -19]
 [  5   5   1 -21 -27 -12  -7]]

Non-zero indices: (array([0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2,
2, 2, 3, 3,
       3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5, 6, 6, 6,
       6, 6, 6, 6, 7, 7, 7, 7, 7, 7, 7, 8, 8, 8, 8, 8, 8, 8]), array([0, 1, 3,
4, 5, 6, 0, 1, 2, 3, 4, 5, 6, 0, 1, 2, 3, 4, 5, 6, 0, 1,
       2, 3, 4, 5, 6, 0, 1, 2, 3, 4, 5, 6, 0, 1, 2, 3, 4, 5, 6, 0, 1, 2,
       3, 4, 5, 6, 0, 1, 2, 3, 4, 5, 6, 0, 1, 2, 3, 4, 5, 6]))

Positive indices: (array([0, 0, 0, 1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 6, 6, 6, 8,
8, 8]), array([2, 4, 6, 5, 2, 3, 2, 3, 6, 0, 1, 3, 4, 2, 0, 2, 4, 0, 1, 2]))


b =
[[-28  -7 -25 -12 -19  -1  -7 -22 -30 -30 -20 -14]
 [-11  -3 -19 -14 -10 -14  -9  -9 -28  -6  -7 -29]
 [ -6 -18 -22 -16 -13 -21 -10 -17  -8  -2  -1  -5]
 [-30 -18  -1 -10  -8 -23  -6 -17  -9 -11  -9 -20]
 [-20  -4 -26  -2  -2 -18 -23 -29  -1  -9 -27 -28]]

Non-zero indices: (array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1,
       1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3,
       3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4]), array([ 0,  1,  2,  3,
4,  5,  6,  7,  8,  9, 10, 11,  0,  1,  2,  3,  4,
        5,  6,  7,  8,  9, 10, 11,  0,  1,  2,  3,  4,  5,  6,  7,  8,  9,
       10, 11,  0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11,  0,  1,  2,
        3,  4,  5,  6,  7,  8,  9, 10, 11]))

Positive indices: (array([], dtype=int64), array([], dtype=int64))
```

6 Question - 06

Two lists are given. Find out how many numbers in the first list are higher or equal than the corresponding numbers in the second list.

n.b. you cannot use for loop for this task.

[13]:
```
x1=[3,7,5,1,8,10,-4,-6]
x2=[10,2,0,4,1,6,-4,-5]

# write your answer below -
```

[14]:
```
def q6_high_equal(x1, x2):
    # list to numpy array
    x1_np = np.array(x1)
    x2_np = np.array(x2)

    out = np.nonzero(x1_np >= x2_np)[0]
    # return out.shape[0]
    print (f"Number of elements: {out.shape[0]}")

# For x1 , x2
q6_high_equal(x1, x2)
```

```
Number of elements: 5
```

7 Question - 07

Iccanobif Numbers

Given a number n. Write a function to find first n Iccanobif Numbers. Iccanobif Numbers are similar to Fibonacci Numbers. The K-th Iccanobif number can be obtained by addition of previous two numbers after reversing their digits.

[15]:
```
def digit_rev(num):
    # d = np.ceil(np.log10(num))
    return int(str(num)[::-1])

def iccanobif(n):
    ic_num = np.zeros((n,) , dtype=int)
    for i in range(n):
        if i == 0:
            ic_num[i] = 0
        elif i == 1:
            ic_num[i] = 1
        else:
            ic_num[i] = digit_rev(ic_num[i-1]) + digit_rev(ic_num[i-2])
    return ic_num
```

[16]:
```
print( iccanobif(12) )
```

```
[  0   1   1   2   3   5   8  13  39 124 514 836]
```

8 Question - 08

Given a list of names of students, add a 'Mr.' in front for male and a 'Ms.' for female.

[17]:
```
name=[('Evaaaaaa','female'),('Wall-e','male'), ('Optimus', '  male')]

#ans= ['Ms. Evaaaaaa', 'Mr.Wall-e']
#write your answer below --
```

[18]:
```
def q7_name(name: list) -> list:
    name = np.array(name)
    n = name.shape[0]
    out = []
    for x in range(n):
        if 'female' in name[x,1]:
            out.append('Ms. ' + name[x,0])
        elif 'male' in name[x,1]:
            out.append('Mr. ' + name[x,0])

    return out

# For name
q7_name(name)
```

[18]:
```
['Ms. Evaaaaaa', 'Mr. Wall-e', 'Mr. Optimus']
```

9 Question - 09

Create a multiplication table from 1 to 15.

- don't use any for loop
- use broadcasting

Hint: np.newaxis

```
[[ 1,  2,  3,  4,  5,  6,  7,  8,  9],
[ 2,  4,  6,  8, 10, 12, 14, 16, 18],
[ 3,  6,  9, 12, 15, 18, 21, 24, 27],
[ 4,  8, 12, 16, 20, 24, 28, 32, 36],
[ 5, 10, 15, 20, 25, 30, 35, 40, 45],
[ 6, 12, 18, 24, 30, 36, 42, 48, 54],
[ 7, 14, 21, 28, 35, 42, 49, 56, 63],
[ 8, 16, 24, 32, 40, 48, 56, 64, 72],
[ 9, 18, 27, 36, 45, 54, 63, 72, 81]]
```

[19]:
```
a= np.arange(1, 16)

# write your code here
```

[20]:
```
def q9_mul_table(n):
    table = np.arange(1,n+1) * np.arange(1,n+1).reshape(-1,1)
    return table

table = q9_mul_table(15)
print(table)
```

```
[[  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15]
 [  2   4   6   8  10  12  14  16  18  20  22  24  26  28  30]
 [  3   6   9  12  15  18  21  24  27  30  33  36  39  42  45]
 [  4   8  12  16  20  24  28  32  36  40  44  48  52  56  60]
 [  5  10  15  20  25  30  35  40  45  50  55  60  65  70  75]
 [  6  12  18  24  30  36  42  48  54  60  66  72  78  84  90]
 [  7  14  21  28  35  42  49  56  63  70  77  84  91  98 105]
 [  8  16  24  32  40  48  56  64  72  80  88  96 104 112 120]
 [  9  18  27  36  45  54  63  72  81  90  99 108 117 126 135]
 [ 10  20  30  40  50  60  70  80  90 100 110 120 130 140 150]
 [ 11  22  33  44  55  66  77  88  99 110 121 132 143 154 165]
 [ 12  24  36  48  60  72  84  96 108 120 132 144 156 168 180]
 [ 13  26  39  52  65  78  91 104 117 130 143 156 169 182 195]
 [ 14  28  42  56  70  84  98 112 126 140 154 168 182 196 210]
 [ 15  30  45  60  75  90 105 120 135 150 165 180 195 210 225]]
```

10 Question - 10

Create a dataframe from the dictionary given. Change the dataframe's last column's name from 'score' to 'rating1'.

[21]:
```
data = {
    'Name': ['Bruce', 'Clark', 'Peter', 'Gandalf'],
    'Identity': ['Batman', 'Superman', 'Spiderman', 'Wizard'],
    'Actors': ['Affleck', 'Henry', 'Maguire', 'Ian'],
    'Age': [44, 27, 22, 50000],
    'Score': [10, 9.5, 9, 9.5]
}
df= pd.DataFrame(data)
df
```

[21]:

	Name	Identity	Actors	Age	Score
0	Bruce	Batman	Affleck	44	10.0
1	Clark	Superman	Henry	27	9.5
2	Peter	Spiderman	Maguire	22	9.0
3	Gandalf	Wizard	Ian	50000	9.5

[22]:
```
# write your code here
def q10_dataframe(data):
    df = pd.DataFrame(data)
```

```
    # change last column name form 'score' to 'rating1'
    df.rename(columns = {'Score':'Rating1'}, inplace = True)
    return df

df = q10_dataframe(data)
df
```

[22]:

	Name	Identity	Actors	Age	Rating1
0	Bruce	Batman	Affleck	44	10.0
1	Clark	Superman	Henry	27	9.5
2	Peter	Spiderman	Maguire	22	9.0
3	Gandalf	Wizard	Ian	50000	9.5

11 Question - 11

Add a new column to your dataframe termed 'Rating-2.

rating-2 = [9, 9, 7.5, 8.5]

[23]:
```
# write your code here
rating2 = [9, 9, 7.5, 8.5]

df['Rating2'] = rating2
df
```

[23]:

	Name	Identity	Actors	Age	Rating1	Rating2
0	Bruce	Batman	Affleck	44	10.0	9.0
1	Clark	Superman	Henry	27	9.5	9.0
2	Peter	Spiderman	Maguire	22	9.0	7.5
3	Gandalf	Wizard	Ian	50000	9.5	8.5

12 Question - 12

Replace the rating1 and rating2 columns with a new column name 'avg_rating'. It will contain the average of those two column values.

[24]:
```
# write your code here

df['avg_rating'] = (df['Rating1'] + df['Rating2']) / 2
df.drop(columns = ['Rating1', 'Rating2'], inplace = True)
df
```

[24]:

	Name	Identity	Actors	Age	avg_rating
0	Bruce	Batman	Affleck	44	9.50
1	Clark	Superman	Henry	27	9.25
2	Peter	Spiderman	Maguire	22	8.25
3	Gandalf	Wizard	Ian	50000	9.00

13 Question - 13

Replace the 3rd row with the following information.

[Celebrimbor, Elf, N/A, 2000, 8]

[25]:
```
# write your code here
new_row = ['Celebrimbor', 'Elf', 'N/A', 2000, 8]
df.loc[2] = new_row
df
```

[25]:

	Name	Identity	Actors	Age	avg_rating
0	Bruce	Batman	Affleck	44	9.50
1	Clark	Superman	Henry	27	9.25
2	Celebrimbor	Elf	N/A	2000	8.00
3	Gandalf	Wizard	Ian	50000	9.00

14 Question - 14

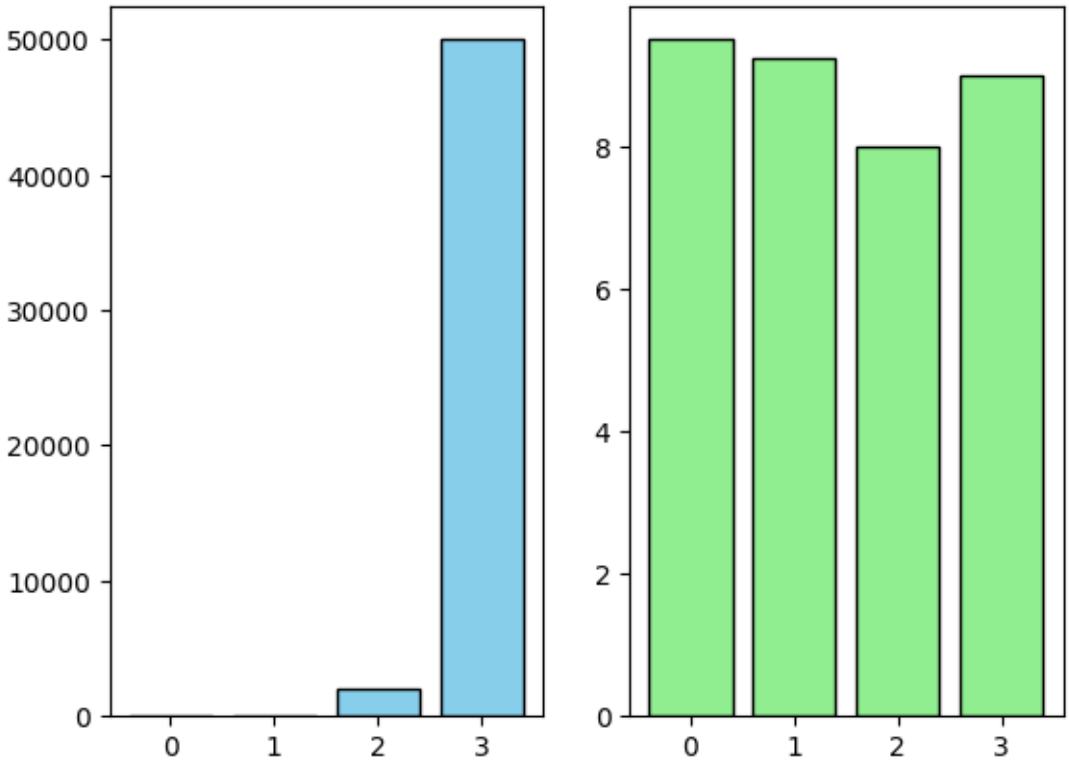
1. Create two separate bar plots from the Age and Rating columns [Use subplot]
2. Create a single bar plot from the Age and Rating columns, both should be placed pair-wise.

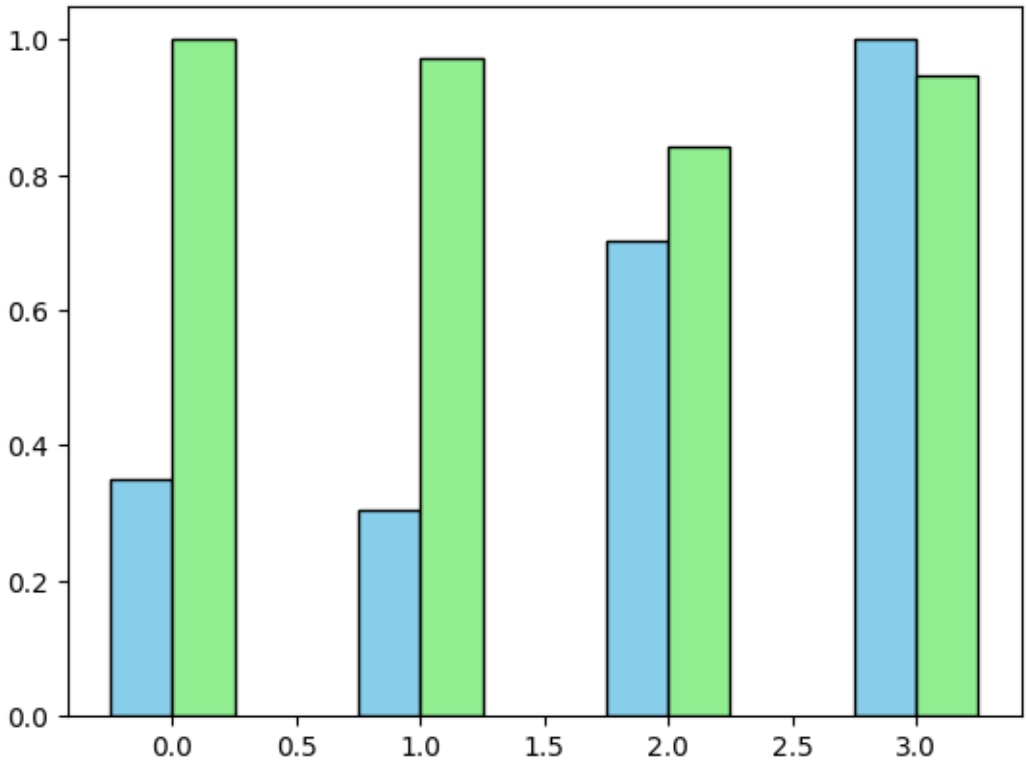
n.b. The scale will create an issue. you may normalize it to view properly.

[26]:
```
import matplotlib.pyplot as plt
```

[27]:
```
# write your code here
def q14_bar_plot(df):
    plt.figure(1)

    plt.subplot(1, 2, 1)
    plt.bar(df.index , df['Age'] , color='skyblue' , edgecolor='black')

    plt.subplot(1, 2, 2)
    plt.bar(df.index , df['avg_rating'] , color='lightgreen' ,␣
 ↪edgecolor='black')

    plt.figure(2)
    # Normalize the data
    age = (np.log2(df['Age'])) / max(np.log2(df['Age']))
    avg_rate = ((df['avg_rating']) / max(df['avg_rating']) )

    plt.bar( df.index-0.125 , age , 0.25 ,  color='skyblue' , edgecolor='black')
    plt.bar( df.index+0.125 , avg_rate , 0.25 , color='lightgreen' ,␣
 ↪edgecolor='black')


    # Display the plot
```

```
    plt.show()

q14_bar_plot(df)
```





15 Question - 15

In Cryptography, Caesar Cipher is one of the simplest form of encryption technique. The concept is very simple. You shift the given message by a fixed degree left or right.

For Instance,

- message= 'fire'
- degree = +3
- encrypted message = 'iluh'

You only shift the alphanumeric characters; not punctuation or whitespace characters.

The characters is so be shifted within the same type i.e. a lowercase should be shifted to a lowercase, a numeric to a numeric.

For example, If you've ('z',+2), it should be shifted to —> 'b'

Hints: * Take a look at dir(str) to see the methods that can be used to identify the type of string. * ord() **provieds the ascii value of a character. * Careful!!, you cannot get vectorized output directly in python. * chr()** converts ascii value back to its character.

[28]:
```
# Solution
def q15_encrypt(message, degree):
```

```
    # msg = np.array(message)
    np_ord = np.vectorize(ord)
    np_cha = np.vectorize(chr)

    msg = np_ord(list(message))

    cap = (msg >= ord('A')) & (msg <= ord('Z'))
    small = (msg >= ord('a')) & (msg <= ord('z'))

    msg[cap | small] += degree

    msg[cap & (msg < ord('A'))] += 26
    msg[small & (msg < ord('a'))] += 26
    msg[cap & (msg > ord('Z'))] -= 26
    msg[small & (msg > ord('z'))] -= 26

    msg = np_cha(msg)
    msg = ''.join(msg)
    return msg
```

[29]:
```
message= 'Radioactive'
degree = +7

# write your answer below -

print(q15_encrypt(message, degree))
```

```
Yhkpvhjapcl
```

[30]:
```
message= '''
Outta order? I'll show you outta order!
You don't know what outta order is, Mr. Trask!
I'd show you but I'm too old; I'm too tired; I'm too fuckin' blind.
If I were the man I was 5 years ago I'd take a FLAME-THROWER to this place!
Outta order. Who the hell you think you're talkin' to?
I've been around, you know? There was a time I could see.
And I have seen boys like these, younger than these, their arms torn out, their␣
 ↪legs ripped off.
But there isn't nothin' like the sight of an amputated spirit; there is no␣
 ↪prosthetic for that.
You think you're merely sendin' this splendid foot-soldier back home to Oregon␣
 ↪with his tail between his legs,
but I say you are executin' his SOUL!!
And why?! Because he's not a Baird man!
Baird men, ya hurt this boy, you're going to be Baird Bums, the lot of ya.
And Harry, Jimmy, Trent, wherever you are out there, **** ***, too!
```

```
'''
degree = -12

# write your answer below -

print(q15_encrypt(message, degree))
```

```
Cihho cfrsf? W'zz gvck mci cihho cfrsf!
Mci rcb'h ybck kvoh cihho cfrsf wg, Af. Hfogy!
W'r gvck mci pih W'a hcc czr; W'a hcc hwfsr; W'a hcc tiqywb' pzwbr.
Wt W ksfs hvs aob W kog 5 msofg ouc W'r hoys o TZOAS-HVFCKSF hc hvwg dzoqs!
Cihho cfrsf. Kvc hvs vszz mci hvwby mci'fs hozywb' hc?
W'js pssb ofcibr, mci ybck? Hvsfs kog o hwas W qcizr gss.
Obr W vojs gssb pcmg zwys hvsgs, mcibusf hvob hvsgs, hvswf ofag hcfb cih, hvswf
zsug fwddsr ctt.
Pih hvsfs wgb'h bchvwb' zwys hvs gwuvh ct ob oadihohsr gdwfwh; hvsfs wg bc
dfcghvshwq tcf hvoh.
Mci hvwby mci'fs asfszm gsbrwb' hvwg gdzsbrwr tcch-gczrwsf poqy vcas hc Cfsucb
kwhv vwg howz pshkssb vwg zsug,
pih W gom mci ofs slsqihwb' vwg GCIZ!!
Obr kvm?! Psqoigs vs'g bch o Powfr aob!
Powfr asb, mo vifh hvwg pcm, mci'fs ucwbu hc ps Powfr Piag, hvs zch ct mo.
Obr Voffm, Xwaam, Hfsbh, kvsfsjsf mci ofs cih hvsfs, **** ***, hcc!
```