

Python Assignment 01

Problem - 1

Python program to solve [Max Split](#)

Code:

```
a=list(map(str,input()))
cnt_L=0
cnt_R=0
cnt=0
b=[]
for i in a:

    if i=='L':
        cnt_L+=1
    else:
        cnt_R+=1
    if cnt_L==cnt_R:
        cnt+=1
        b.append(''.join((a[:cnt_L+cnt_R])))
        a=a[cnt_L + cnt_R:]
        cnt_L=0
        cnt_R=0
print(cnt)
for s in b:
    print(s)
```

Problem - 2

program to solve [Good Sequence](#)

Code:

```
n=int(input())

a=list(map(int,input().split()))

from collections import Counter

b=Counter(a)
re=0
for k,c in b.items():
    if k!=c:
        re+=(min(c,abs(k-c)))

print(re)
```

Problem - 3

a

Difference between List and Dictionary of Python.

Answer:

(1)

- Lists are ordered collections but
- Dictionaries are unordered collections of key-value pairs.

(2)

- List is defined using square brackets [].
Example: list_1 = [1, 2, 3, "j"].
- On the other hand, Dictionary is defined using curly braces {}.
Example: my_dict = {"key1": 42, "key2": "value"}.

(3)

- Elements in a list are accessed using indices but
- Elements in a dictionary are accessed using keys rather than indices.

(4)

- List Allows duplicates elements but
- Dictionaries does not allows duplicate element.

(5)

- List object is created using list()function but
- Dictionaries use dict() function.

b

*args and **kwargs of Python with proper examples.

Answer:

***args (Arbitrary Arguments):**

It allows a function to accept any number of positional arguments. The asterisk (*) before the parameter name args allows to pass any number of arguments when calling the function.

Example:

```
def j_function(*args):  
    for arg in args:  
        print(arg)
```

```
j_function(1, 2, 3, 4)
```

****kwargs (Keyword Arguments):**

It allows a function to accept any number of keyword arguments. The double asterisk (**) before the parameter name kwargs allows to pass any number of keyword arguments when calling the function.

Example:

```
def j_info(**kwargs):  
    for key, value in kwargs.items():  
        print(f'{key}: {value}')
```

j_info(name="Jahid", age=23, city="Ctg", occupation="Engineering Student")

Problem - 4

program to solve [Minimize Number](#)

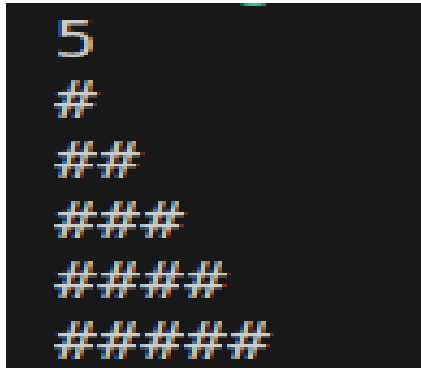
Code:

```
n= int(input())
a = list(map(int, input().split()))
op= 0
while all(x%2==0 for x in a):
    a = [x // 2 for x in a]
    op+= 1
print(op)
```

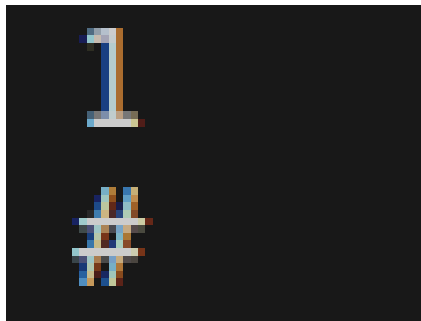
Problem - 5

→ Take a number from the user and draw a pyramid using PyAutoGUI

Sample :



```
5
#
##
###
####
#####
```

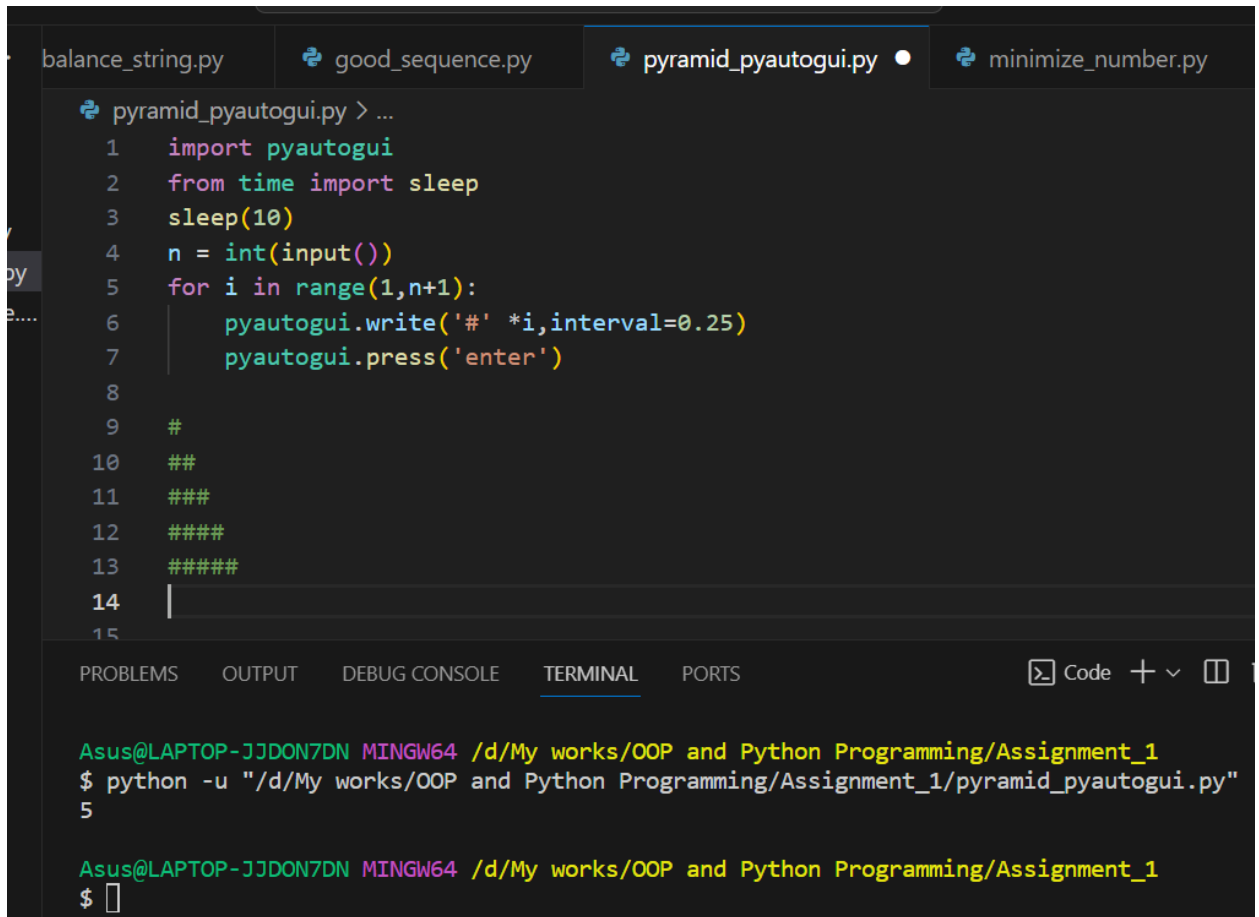


```
1
#
```

Code:

```
import pyautogui
from time import sleep
sleep(10)
n = int(input())
for i in range(1,n+1):
    pyautogui.write('#' *i,interval=0.25)
    pyautogui.press('enter')
```

Output:



The image shows a code editor with four tabs: `balance_string.py`, `good_sequence.py`, `pyramid_pyautogui.py` (selected), and `minimize_number.py`. The `pyramid_pyautogui.py` tab contains the following Python code:

```
1 import pyautogui
2 from time import sleep
3 sleep(10)
4 n = int(input())
5 for i in range(1,n+1):
6     pyautogui.write('#' *i, interval=0.25)
7     pyautogui.press('enter')
8
9 #
10 ##
11 ###
12 ####
13 #####
14 |
15
```

Below the code editor is a terminal window with the following output:

```
Asus@LAPTOP-JJDON7DN MINGW64 /d/My works/OOP and Python Programming/Assignment_1
$ python -u "/d/My works/OOP and Python Programming/Assignment_1/pyramid_pyautogui.py"
5

Asus@LAPTOP-JJDON7DN MINGW64 /d/My works/OOP and Python Programming/Assignment_1
$
```