

## Contents

SET 1.....	2
1. Given a list of elements all elements are equal except one. Write a code to find the odd man out(Stray number).....	2
2. In the given list of elements, find the element which is close to its mean.....	3
3. Find the average speed of vehicle, given the distance travelled for fixed time intervals, e.g. [0,0.1,0.25,0.45,0.55,0.7,0.9,1.0].....	3
4. Find the number of people in a bus, given the data of people onboarding and alighting at each station. ....	4
5. Find the missing number given the original and modified one. ....	4
6. Find the difference between two lowest numbers in the list.....	4
7. In a given list, count number of elements smaller than mean .....	5
SET 2.....	6
1. Correct the malformed time string, for e.g. "5:70:65" to "6:11:05" .....	6
2. Correct the malformed date string, for e.g. "45/8/2018" to "14/9/2018" .....	6
3. Convert IP address to integer and vice versa.....	7
4. Check whether the given string is isogram or not. ....	8
5. Given a string, find the Mexican wave.....	8
6. Given a number, find the largest number by deleting a single digit (Order will remain same) ...	9
7. Given a number, find the largest number by shuffling it.....	10
8. Compute word frequency in given message.....	10
9. Convert RGB to Hex and vice versa.....	11
10. Generate accumulated strings, e.g. abcd == > A-Bb-Ccc-Dddd .....	11

# SET 1

1. Given a list of elements all elements are equal except one. Write a code to find the odd man out(Stray number)

```
mylist = [4, 2, 2, 2, 2]

a = mylist[0]
k = mylist[1]

if a != k:
    p = mylist.count(a)
    q = mylist.count(k)
    if p > q:
        print("The odd element is", k)
    else:
        print("The odd element is", a)
else:
    for i in range(1, len(mylist)):
        if a == mylist[i]:
            e = mylist[i]
        else:
            f = mylist[i]
            m = i + 1
            b = mylist[m]
            c = mylist.count(f)
            d = mylist.count(b)
            if c > d:
                print("The odd element is", mylist[m])
            else:
                print("The odd element is", mylist[i])
```

- *Output:*

```
The odd element is 4

Process finished with exit code 0
```

2. In the given list of elements, find the element which is close to its mean.

```
list_1 = [1, 2, 3, 4, 5, 6, 7]
t = 0
new_list = []
for i in list_1:
    t += i
mean = t / len(list_1)
# print(mean)
for i in range(0, len(list_1)):
    new_list.append(mean - list_1[i])
    if new_list[i] < 0:
        new_list[i] = new_list[i] * (-1)
min = new_list[0]
for i in range(0, len(new_list)):
    if min > new_list[i]:
        min = new_list[i]
for i in range(0, len(new_list)):
    if min == new_list[i]:
        print(list_1[i])
        break
```

- *Output:*

```
4

Process finished with exit code 0
```

3. Find the average speed of vehicle, given the distance travelled for fixed time intervals, e.g. [0,0.1,0.25,0.45,0.55,0.7,0.9,1.0]

```
distance = [0, 0.1, 0.25, 0.45, 0.55, 0.7, 0.9, 1.0]
time = 1
total = 0
for i in distance:
    total += i
avg_speed = total/len(distance)
```

```
print("The average speed is \n", avg_speed)
```

- *Output:*

```
The average speed is
0.49374999999999997

Process finished with exit code 0
```

4. Find the number of people in a bus, given the data of people onboarding and alighting at each station.

```
pas_onboard = [30, 23, 35, 14, 45, 26, 19]
pas_alight = [33, 25, 23, 44, 21, 16, 11]
pas = 0
for x in range(len(pas_onboard)):
    pas += pas_onboard[x] - pas_alight[x]

print(pas)
```

- *Output:*

```
19

Process finished with exit code 0
```

5. Find the missing number given the original and modified one.

```
list_p = [12, 23, 14, 36, 27, 18, 20]
mod_list = [12, 23, 14, 27, 18, 20]
for i in range(0, len(list_p)):
    if list_p[i] in mod_list:
        continue
    else:
        print("Missing element is", list_p[i])
```

- *Output:*

```
Missing element is 36

Process finished with exit code 0
```

6. Find the difference between two lowest numbers in the list.

```
my_list = [4, 8, 9, 3, 6, 1]
new_list = []
while my_list:
    mini = my_list[0]
```

```
for x in my_list:
    if x < mini:
        mini = x
new_list.append(mini)
my_list.remove(mini)
print("The difference between two lowest numbers is", abs(new_list[1] - new_list[0]))
```

- *Output:*

```
The difference between two lowest numbers is 2

Process finished with exit code 0
```

7. In a given list, count number of elements smaller than mean

```
my_list = [12, 13, 16, 17, 19, 20]
new_list = []
a = 0
for i in my_list:
    a = a + i
mean = a/len(my_list)

for x in range(0, len(my_list)):
    b = my_list[x]
    if b < mean:
        new_list.append(b)

print("The count of number of elements smaller than the mean is", len(new_list))
```

- *Output:*

```
The count of number of elements smaller than the mean is 3

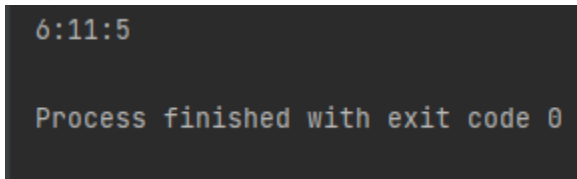
Process finished with exit code 0
```

## SET 2

1. Correct the malformed time string, for e.g. "5:70:65" to "6:11:05"

```
input_time = "5:70:65"
list_time = input_time.split(':')
hh = int(list_time[0])
mm = int(list_time[1])
ss = int(list_time[2])
correct_time = ""
if ss >= 60:
    extra_min = ss // 60
    mm = mm + extra_min
    ss = ss % 60
if mm >= 60:
    extra_hour = mm // 60
    hh = hh + extra_hour
    mm = mm % 60
if hh >= 24:
    hh = hh % 24
correct_time = str(hh) + ":" + str(mm) + ":" + str(ss)
print(correct_time)
```

- *Output:*



```
6:11:5
Process finished with exit code 0
```

2. Correct the malformed date string, for e.g. "45/8/2018" to "14/9/2018"

```
months = {1: 31, 2: 28, 3: 31, 4: 30, 5: 31, 6: 30, 7: 31, 8: 31, 9: 30, 10: 31, 11: 30, 12: 31}
input_date = input("Enter the date (dd/mm/yyyy): \n")
list_date = input_date.split('/')
dd = int(list_date[0])
mm = int(list_date[1])
yyyy = int(list_date[2])
newDate = ""
if (yyyy % 4) == 0:
    if (yyyy % 100) == 0:
        if (yyyy % 400) == 0:
            months[2] = 29
```

```

        else:
            months[2] = 28
    else:
        months[2] = 29
    else:
        months[2] = 28
    c1 = dd // months[int(mm)]
    rem1 = dd % months[int(mm)]
    mm += c1
    dd = rem1
    c2 = mm // 12
    rem2 = mm % 12
    yyyy += c2
    mm = rem2
    print('The correct date is', dd, '/', mm, '/', yyyy)

```

- *Output:*

```

Enter the date (dd/mm/yyyy):
45/8/2018
The correct date is 14 / 9 / 2018

Process finished with exit code 0

```

### 3. Convert IP address to integer and vice versa

```

ip = input("Enter the ip address (a.b.c.d format): \n")
p = 0
int_value = 0
for i in ip.split('.')[::-1]:
    int_value += int(i) * 256**p
    p += 1
print(int_value)

ip_address = ""
for i in range(3, -1, -1):
    num = int_value // (256**i)
    ip_address = ip_address + str(num)
    int_value = int_value % (256**i)
    if i > 0:
        ip_address = ip_address + '.'
print(ip_address)

```

- *Output:*

```
Enter the ip address (a.b.c.d format):  
192.168.0.1  
3232235521  
192.168.0.1  
  
Process finished with exit code 0
```

4. Check whether the given string is isogram or not.

```
word_in = input("Enter string \n")  
word = word_in.lower()  
# print(word)  
a = 0  
char_list = []  
for char in word:  
    if char.isalpha():  
        if char in char_list:  
            a += 1  
        char_list.append(char)  
if a >= 1:  
    print("Input string is not an isogram")  
else:  
    print("Input string is an isogram")
```

- *Output:*

```
Enter string  
Python Project  
Input string is not an isogram  
  
Process finished with exit code 0
```

5. Given a string, find the Mexican wave.

```
word = 'alphabet'  
new = []  
for i, val in enumerate(word[:]):  
    up = word[i].upper()  
    c = word[:i] + up + word[i+1:]  
    new.append(c)  
print(new)
```



- *Output:*

```
Enter string
Prachi
['Prachi', 'PRachi', 'PrAchi', 'PraChi', 'PracHi', 'PrachI']

Process finished with exit code 0
|
```

6. Given a number, find the largest number by deleting a single digit (Order will remain same)

```
num = input("Enter number \n")
res = [int(x) for x in str(num)]
li = [int(x) for x in str(num)]
# print(li)
new_list = []
while li:
    mini = li[0]
    for x in li:
        if x < mini:
            mini = x
    new_list.append(mini)
    li.remove(mini)
# print(new_list)
a = new_list[0]
# print(a)
# print(res)
for i in range(0, len(res)):
    if res[i] == a:
        b = i
        break
res.remove(res[b])
# print(res)
strings = [str(integer) for integer in res]
a_string = "".join(strings)
an_integer = int(a_string)
print("The largest number deleting single digit is", an_integer)
```

- *Output:*

```
Enter number
13571987
The largest number deleting single digit is 3571987

Process finished with exit code 0
```

7. Given a number, find the largest number by shuffling it.

```
number = 19216804
new = []
s = str(number)
new[:0] = s
new_list = []
while new:
    mini = new[0]
    for x in new:
        if x > mini:
            mini = x
    new_list.append(mini)
    new.remove(mini)

s1 = ""

for x in new_list:
    s1 += x

print(int(s1))
```

- *Output:*

```
98642110

Process finished with exit code 0
```

8. Compute word frequency in given message

```
str = "she laughs she dances she plays she eats"
str = str.split()
str2 = []

for i in str:
    if i not in str2:
        str2.append(i)
```

```
for i in range(0, len(str2)):
    print('Frequency of', str2[i], 'is :', str.count(str2[i]))
```

- *Output:*

```
Frequency of she is : 4
Frequency of laughs is : 1
Frequency of dances is : 1
Frequency of plays is : 1
Frequency of eats is : 1

Process finished with exit code 0
```

## 9. Convert RGB to Hex and vice versa

```
#Hex to RGB
value = "FF65BA"
print("Hex input:", value)
value = value.lstrip('#')
print('RGB =', tuple(int(value[i:i+2], 16) for i in (0, 2, 4)))
#RGB to Hex
rgb = (255, 230, 50)
print("RGB input:", rgb)
print('Hex = %02x%02x%02x' % rgb)
```

- *Output:*

```
Hex input: FF65BA
RGB = (255, 101, 186)
RGB input: (255, 230, 50)
Hex = ffe632

Process finished with exit code 0
```

## 10. Generate accumulated strings, e.g. abcd ==> A-Bb-Ccc-Dddd

```
word = input("Enter string \n")
accum_str = ""
r = range(len(word))
for n in r:
    ch = word[n]
    accum_str += '-' + ch.upper() + ch.lower() * n
print(accum_str[1:])
```

- *Output:*

```
Enter string
acegik
A-Cc-Eee-Gggg-Iiiii-Kkkkkk

Process finished with exit code 0
```