Extra Task :

1)

```
x = [100, 200, 300, 400, 500, [1, 2, 3, 4, 5, [10, 20, 30, 40, 50], 6, 7, 8,
9], 600, 700, 800]
print(x[5][:4])
print(x[-3:-1])
print(x[::2])
y = x[::-1]
print([y])
print(x[5][5][0])
y = x[0:0:]
print(y)
```

output:

```
(venv) C:\Users\Vaishnavi\PycharmProjects\pythonProject2>python task.py
[1, 2, 3, 4]
[600, 700]
[100, 300, 500, 600, 800]
[[800, 700, 600, [1, 2, 3, 4, 5, [10, 20, 30, 40, 50], 6, 7, 8, 9], 500, 400, 300,
 200, 100]]
10
[]
```

2)

```
x = range(1,1000,1)
print(x)
print(type(x))
```

output:

```
(venv) C:\Users\Vaishnavi\PycharmProjects\pythonProject2>python task.py
range(1, 1000)
<class 'range'>
```

```
x = xrange(1,1000,1)
print(x)
print(type(x))
```

output:

```
(venv) C:\Users\Vaishnavi\PycharmProjects\pythonProject2>python task.py
Traceback (most recent call last):
  File "C:\Users\Vaishnavi\PycharmProjects\pythonProject2\task.py", line 1, in <m
dule>
    x = xrange(1,1000,1)
NameError: name 'xrange' is not defined
```

3)

Lists are mutable while tuples are immutable.
When the iterations are applied to the list objects, the processing time is more. On the contrary, tuple objects iterate in a fast manner.
Tuple data type is appropriate for accessing the elements. As against, the list is better for performing operations, such as insertion and deletion.
If the user wants to utilize the sequence as a dictionary key, a tuple is a better choice because dictionary keys are immutable.
Lists consume more memory as compared to the tuple.
Lists have several built-in methods while tuple does no have must built-in methods.

4)

```
res = []
def divi():
    for x in range(1,101):
        if x%3==0 and x%2 == 0:
            res.append(x)
    print(res)


divi()
```

output:

```
Terminal:  Local  +

(venv) C:\Users\Vaishnavi\PycharmProjects\pythonProject2>python task.py
[6, 12, 18, 24, 30, 36, 42, 48, 54, 60, 66, 72, 78, 84, 90, 96]
```

5)

```
str = "Vowel"
y = str[::-1]
print("Given String:", str)
print("Reversed String:", y)

for index, element in enumerate(str):
    if element in ('a', 'e', 'i', 'o', 'u'):
        print("{} element in index {}".format(element, index))
```

output:

```
(venv) C:\Users\Vaishnavi\PycharmProjects\pythonProject2>python task.py
Given String: Vowel
Reversed String: lewoV
o element in index 1
e element in index 3
```

6)

```
str = "hello my name is abcde"

for i in str.split(" "):
    if len(i) % 2 == 0:
        print(i)
```

output:

```
(venv) C:\Users\Vaishnavi\PycharmProjects\pythonProject2>python task.py
my
name
is
```

7)

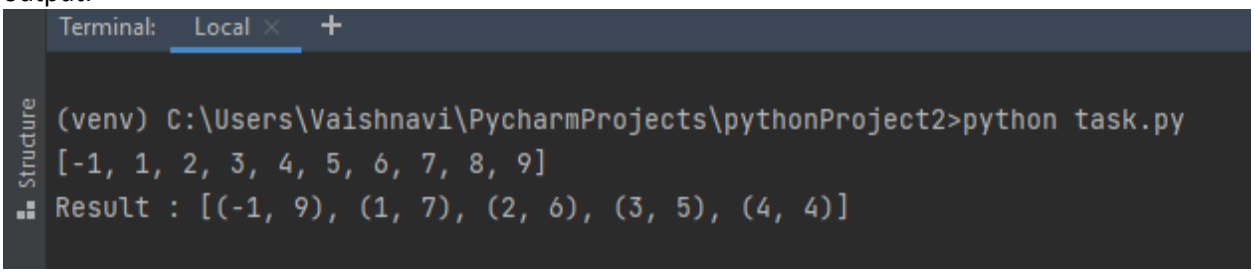
```
x.sort()
print(x)
target = 8
res = []

for i in range(0, len(x)):
    for j in range(i, len(x)):
        sum1 = x[i] + x[j]

        if sum1 == target:
            temp = x[i], x[j]
            res.append(temp)

print("Result :", res )
```

output:



8)

```
even_list = []
odd_list = []
```

```
for i in range(0, 10):
    inp = eval(input("Enter a number in the range of 1,50: "))
    if(inp%2 == 0):
        even_list.append(inp)
        print(even_list)
        if (len(even_list) == 5):
            break
    else:
        odd_list.append(inp)
        print(odd_list)
        if (len(odd_list) == 5):
            break

sum_odd = sum(odd_list)
print("Sum of odd_list {}:".format(odd_list), sum_odd)

sum_evn = sum(even_list)
print("Sum of even_list {}:".format(even_list), sum_evn)

print("Maximum of the list is: ", max(sum_evn,sum_odd))
```

output:

```
(venv) C:\Users\Vaishnavi\PycharmProjects\pythonProject2>python task.py
Enter a number in the range of 1,50: 4
[4]
Enter a number in the range of 1,50: 45
[45]
Enter a number in the range of 1,50: 24
[4, 24]
Enter a number in the range of 1,50: 23
[45, 23]
Enter a number in the range of 1,50: 44
[4, 24, 44]
Enter a number in the range of 1,50: 13
[45, 23, 13]
Enter a number in the range of 1,50: 26
[4, 24, 44, 26]
Enter a number in the range of 1,50: 25
[45, 23, 13, 25]
Enter a number in the range of 1,50: 14
[4, 24, 44, 26, 14]
Sum of odd_list [45, 23, 13, 25]: 106
Sum of even_list [4, 24, 44, 26, 14]: 112
Maximum of the list is:  112
```

9)

```
str =" 12abcbacbaba344ab"
dict = {}
for ch in str:
    if ch.isalpha() and dict.__contains__(ch):
        dict[ch] = dict.get(ch)+1
    elif ch.isalpha():
        dict[ch] = 1

print(dict)
```

output:

```
(venv) C:\Users\Vaishnavi\PycharmProjects\pythonProject2>python task.py
{'a': 5, 'b': 5, 'c': 2}
```

10)

```
t = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
res = []
for item in t:
    if item % 2 == 0:
        res.append(item)

print(tuple(res))
```

output:

```
(venv) C:\Users\Vaishnavi\PycharmProjects\pythonProject2>python task.py
(2, 4, 6, 8, 10)
```