

```
'''Write a Python program to print all even numbers from a given list of numbers in the same order. Stop printing if any number that comes after 237 in the sequence is encountered.
```

Sample

numbers list :

```
numbers = [ 386, 462, 47, 418, 907, 344, 236, 375, 823, 566, 597, 978, 328, 615, 953, 345,  
399, 162, 758, 219, 237, 412, 566, 731, 210, 912, 216, 244, 896, 101, 867, 355, 430 ]
```

expected output:

386 462 418 344 236 566 978 328 162 758

三

```
numbers = [ 386, 462, 47, 418, 907, 344, 236, 375, 823, 566, 597, 978, 328, 615, 953, 345,  
399, 162, 758, 219, 237, 412, 566, 731, 210, 912, 216, 244, 896, 101, 867, 355, 430 ]
```

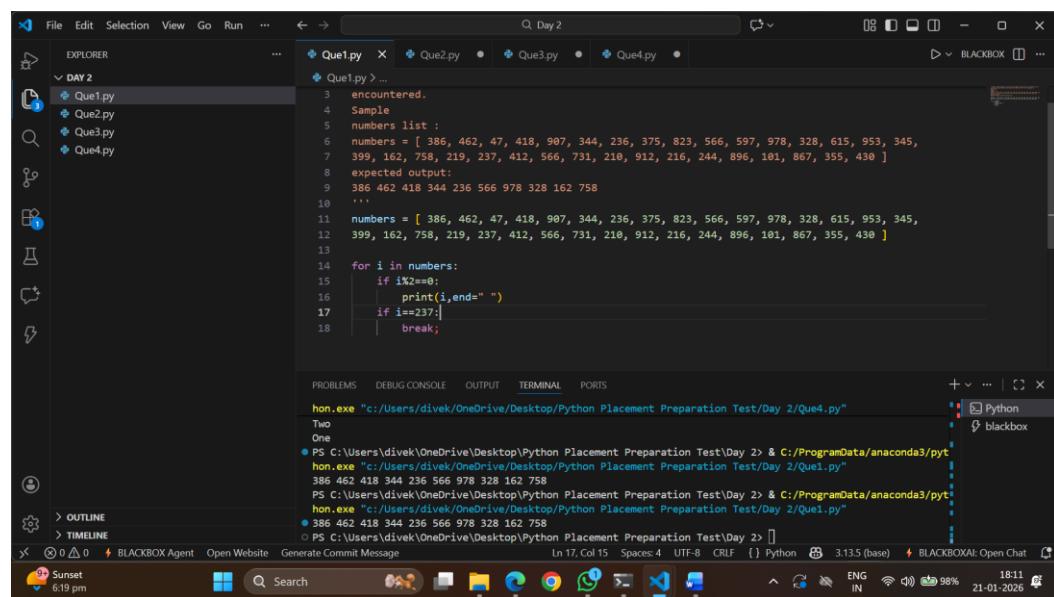
```
for i in numbers:
```

```
if i%2==0:
```

```
print(i,end=" ")
```

```
if i==237:
```

break;



2. import os

```
current_file_path = os.path.abspath(__file__)
current_file_name = os.path.basename(__file__)

print("Path:", current_file_path)
print("Name:", current_file_name)
```

The screenshot shows a code editor interface with a dark theme. The left sidebar has icons for File, Edit, Selection, View, Go, Run, etc. The Explorer panel shows a folder structure for 'DAY 2' containing files 'Que1.py', 'Que2.py', 'Que3.py', and 'Que4.py'. The 'Que2.py' file is selected and its content is displayed in the main editor area:

```
1 import os
2
3
4 current_file_path = os.path.abspath(__file__)
5 current_file_name = os.path.basename(__file__)
6
7 print("Path:", current_file_path)
8 print("Name:", current_file_name)
9
```

The terminal below shows command-line history:

```
PS C:\Users\divek\OneDrive\Desktop\Python Placement Preparation Test\Day 2> & C:/ProgramData/anaconda3/python.exe "c:/Users/divek/OneDrive/Desktop/Python Placement Preparation Test/Day 2/Que1.py"
PS C:\Users\divek\OneDrive\Desktop\Python Placement Preparation Test\Day 2> & C:/ProgramData/anaconda3/python.exe "c:/Users/divek/OneDrive/Desktop/Python Placement Preparation Test/Day 2/Que1.py"
 386 462 418 344 236 566 978 328 162 758
PS C:\Users\divek\OneDrive\Desktop\Python Placement Preparation Test\Day 2> & C:/ProgramData/anaconda3/python.exe "c:/Users/divek/OneDrive/Desktop/Python Placement Preparation Test/Day 2/Que2.py"
Path: c:/Users/divek/OneDrive/Desktop/Python Placement Preparation Test\Day 2\Que2.py
Name: Que2.py
PS C:\Users\divek\OneDrive\Desktop\Python Placement Preparation Test\Day 2> []
```

The bottom status bar shows system information: 27°C, Mostly cloudy, BLACKBOX Agent, Open Website, Generate Commit Message, In 3, Col 1, Spaces: 4, UTF-8, CR/LF, Python 3.13.5 (base), ENG IN, 98%, 21-01-2026, 18:13.

3'''

pattern

```
1  
212  
32123  
4321234  
543212345  
'''
```

for i in range(6):

```
    for j in range(6-i,1,-1):
```

```
        print(" ",end="")
```

```
        for k in range(i,0,-1):
```

```
            print(k,end='')
```

```
            for l in range(2,i+1,1):
```

```
                print(l,end='')
```

```
        print()
```

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer View:** Shows a folder named "DAY 2" containing files: Que1.py, Que2.py, Que3.py (selected), and Que4.py.
- Code Editor:** Displays the content of Que3.py. The code uses nested loops to print a pattern of numbers. It starts with three blank lines, followed by a single '1', then '212', then '32123', then '4321234', then '543212345', and finally three blank lines. Below this, it contains a nested loop structure that prints a sequence of numbers from 1 to 543212345.
- Terminal:** Shows the command "hon.exe" being run with arguments "c:/Users/divek/OneDrive/Desktop/Python Placement Preparation Test/Day 2/Que3.py". The output of the program is displayed, showing the pattern of numbers.
- Status Bar:** Shows the current weather as "27°C Mostly cloudy", system status like "ENG IN", battery level at "98%", and the date and time "21-01-2026 18:14".

4. ""

Write a code to accept a number & print its digits in words .

Ex: 321

Three

Two

One

...

```
dic = {
```

```
    '0': 'Zero',
```

```
    '1': 'One',
```

```
    '2': 'Two',
```

```
    '3': 'Three',
```

```
    '4': 'Four',
```

```
    '5': 'Five',
```

```
    '6': 'Six',
```

```
    '7': 'Seven',
```

```
    '8': 'Eight',
```

```
    '9': 'Nine'
```

```
}
```

```
num = int(input("Enter a number: "))
```

```
for i in str(num):
```

```
    print(dic[i])
```

```
1  """
2  Write a code to accept a number & print its digits in words .
3  Ex: 321
4  Three
5  Two
6  One
7  ...
8
9  dic = {
10     '0': 'Zero',
11     '1': 'One',
12     '2': 'Two',
13     '3': 'Three',
14     '4': 'Four',
15     '5': 'Five',
16     '6': 'Six',
17     '7': 'Seven',
18     '8': 'Eight',
19     '9': 'Nine'
}
20
21 num = int(input("Enter a number: "))
22
23 for i in str(num):
24     print(dic[i])
```

The screenshot shows a code editor window with a file named "Que4.py". The code defines a dictionary "dic" that maps integers from 0 to 9 to their corresponding words. It then prompts the user to enter a number, converts it to a string, and iterates over each character (digit) to print its word representation from the dictionary. When run in a terminal, it correctly prints "Three" and "Two" for the input "321".