

Python

"Q.1) Write a Python function to check whether a string is a pangram or not.
Note : Pangrams are words or sentences containing every letter of the alphabet at least once.
For example : "The quick brown fox jumps over the lazy dog" ""

```
def is_pangram(s):
    s = s.lower()
    for ch in range(ord('a'), ord('z') + 1):
        if chr(ch) not in s:
            return False
    return True

text = "The quick brown fox jumps over the lazy dog"
print(is_pangram(text))
```

Output:

```
C:\Users\gadha\AppData\Local\Programs\Python\Python313\python.exe E:\ML\Test_5.py
True

Process finished with exit code 0
```

"Q.2) Write a Python program to calculate the sum of the digits in an integer. "

```
interger = input("Enter an integer: ")
mylist = []
for i in interger:
    mylist.append(int(i))
for i in mylist:
    total = sum(mylist)
print("Sum of total digit is :",total)
```

Output:

```
C:\Users\gadha\AppData\Local\Programs\Python\Python313\python.exe E:\ML\Test_5.py
Enter an integer: 456
Sum of total digit is : 15

Process finished with exit code 0
```

'''Q.3) Write a Python program to sort three integers without using conditional statements and loops. [u can use built in functions for this] '''

```
mylist=[]
for i in range(3):
    num = int(input("Enter a number: "))
    mylist.append(num)
    mylist.sort()
print(mylist)
```

Output:

```
Q | :
↑ C:\Users\gadha\AppData\Local\Programs\Python\Python313\python.exe E:\ML\Test_5.py
↓ Enter a number: 11
→ Enter a number: 22
→ Enter a number: 33
> [11, 22, 33]
> Test_5.py
```

'''Q.4) Write a Python function to check whether a number is perfect or not.

According to Wikipedia : In number theory, a perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself (also known as its aliquot sum). Equivalently, a perfect number is a number that is half the sum of all of its positive divisors (including itself).

Example : The first perfect number is 6, because 1, 2, and 3 are its proper positive divisors, and $1 + 2 + 3 = 6$. Equivalently, the number 6 is equal to half the sum of all its positive divisors: $(1 + 2 + 3 + 6) / 2 = 6$. The next perfect number is $28 = 1 + 2 + 4 + 7 + 14$. This is followed by the perfect numbers 496 and 8128. "

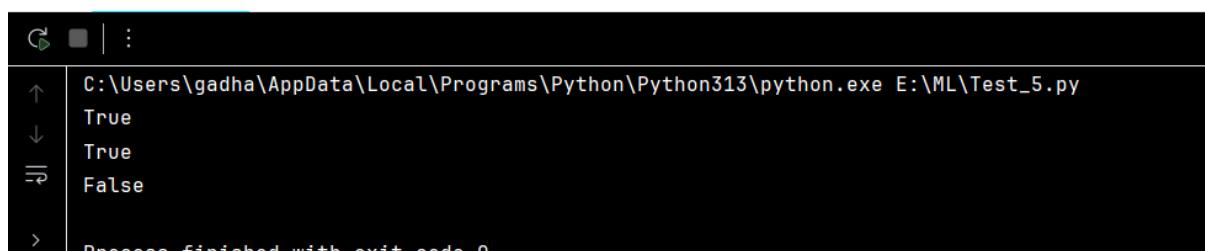
```
def is_perfect(num):
    if num <= 0:
        return False

    sum_divisors = 0
    for i in range(1, num):
        if num % i == 0:
            sum_divisors += i

    return sum_divisors == num

print(is_perfect(6))
print(is_perfect(28))
print(is_perfect(12))
```

Output:



```
C:\Users\gadha\AppData\Local\Programs\Python\Python313\python.exe E:\ML\Test_5.py
True
True
False
> Process finished with exit code 0
```