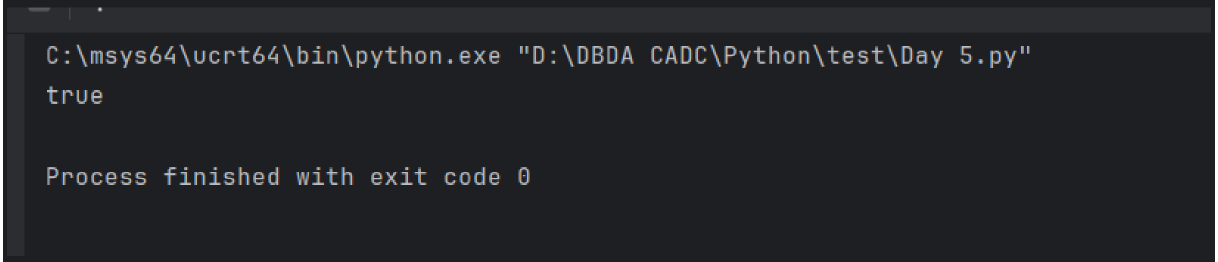
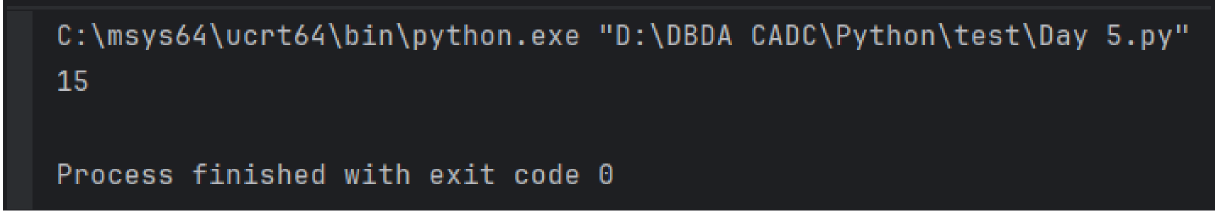
```
# Q.1) Write a Python function to check whether a string is a pangram or
not.
# Note : Pangrams are words or sentences containing every letter of the
alphabet at least once.
# For example : "The quick brown fox jumps over the lazy dog"
# from string import ascii_lowercase
#
# def checkPangram(s):
#     for ch in ascii_lowercase:
#         found = False
#
#         for i in range(len(s)):
#             if ch == (s[i].lower()):
#                 found = True
#                 break
#
#         if found == False:
#             return False
#     return True
#
#
# if __name__ == "__main__":
#     s = "The quick brown fox jumps over the lazy dog"
#     if checkPangram(s) == True:
#         print("true")
#     else:
#         print("false")
```



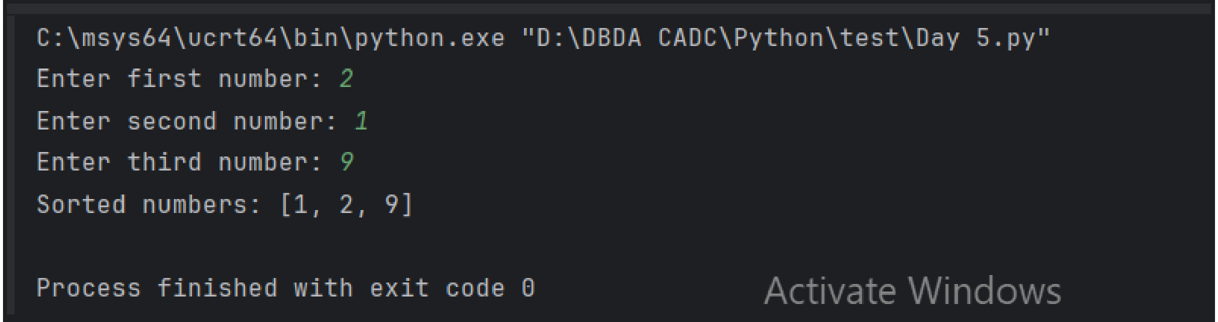
```
C:\msys64\ucrt64\bin\python.exe "D:\DBDA CADC\Python\test\Day 5.py"
true

Process finished with exit code 0
```

```
# Q.2) Write a Python program to calculate the sum of the digits in an
integer.
# n = 12345
# sum = 0
#
# while n > 0:
#     sum += n % 10
#     n //= 10
#
# print(sum)
```



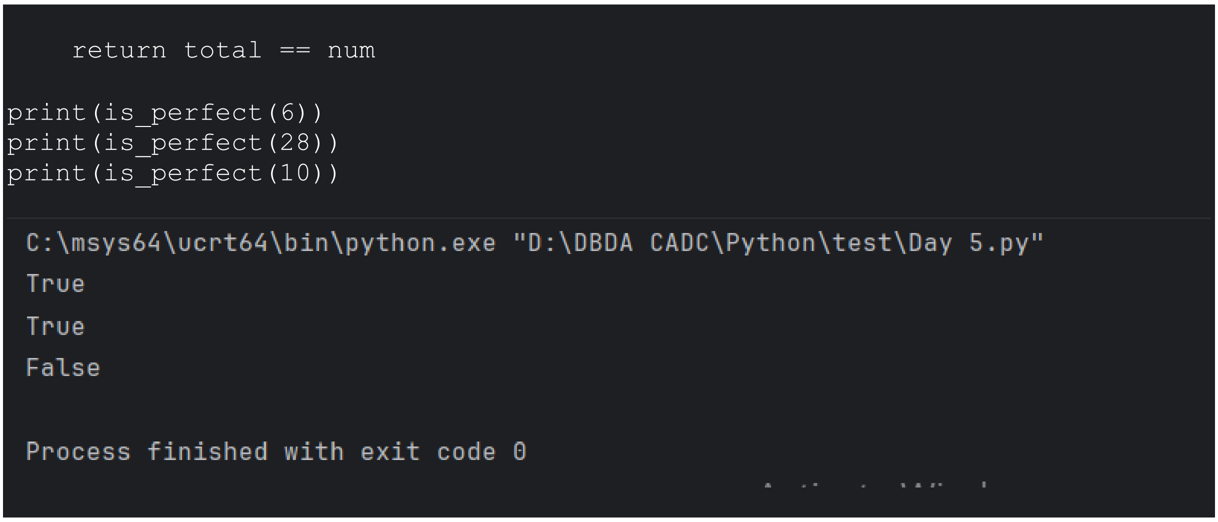
```
C:\msys64\ucrt64\bin\python.exe "D:\DBDA CADC\Python\test\Day 5.py"
15

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 ]

# Input three integers
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))
c = int(input("Enter third number: "))

# Sort using built-in function
sorted_numbers = sorted([a, b, c])

# Output
print("Sorted numbers:", sorted_numbers)
```



```
C:\msys64\ucrt64\bin\python.exe "D:\DBDA CADC\Python\test\Day 5.py"
Enter first number: 2
Enter second number: 1
Enter third number: 9
Sorted numbers: [1, 2, 9]

Process finished with exit code 0
```

```
---------------------------------------------------------------------------

# 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 i              s 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 <= 1:
        return False

    total = 1  # 1 is always a divisor

    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            total += i
            if i != num // i:
                total += num // i
```



```
    return total == num

print(is_perfect(6))
print(is_perfect(28))
print(is_perfect(10))
```

```
C:\msys64\ucrt64\bin\python.exe "D:\DBDA CADC\Python\test\Day 5.py"
True
True
False

Process finished with exit code 0
```