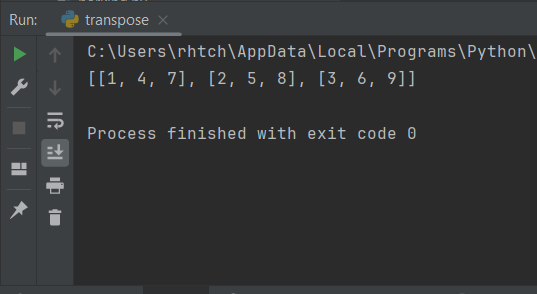
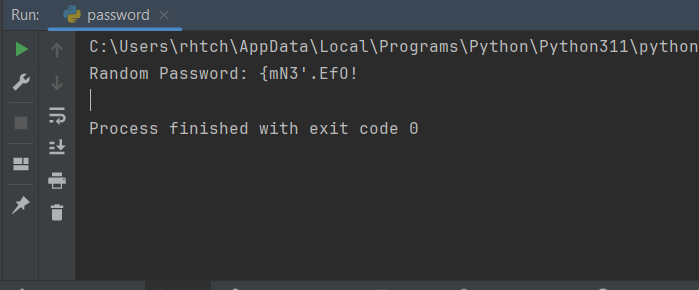
**Write a Python function that takes a 2D list (matrix) and returns its transpose.**

def transpose(matrix):  
 transposed = []  
 for i in range(len(matrix[0])):  
 transposed\_row = []  
 for j in range(len(matrix)):  
 transposed\_row.append(matrix[j][i])  
 transposed.append(transposed\_row)  
 return transposed  
  
matrix = [  
 [1,2,3],  
 [4,5,6],  
 [7,8,9]  
]  
print(transpose(matrix))



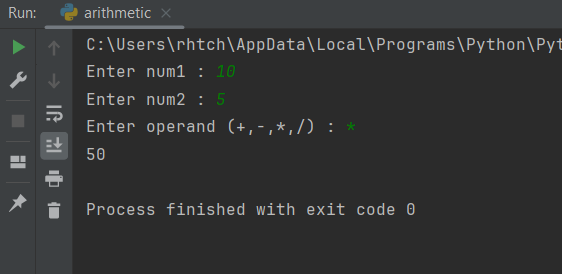
**Write a Python function that generates a random password. The password should contain a mix of uppercase letters, lowercase letters, digits, and special characters.**

import random  
  
def generate\_random\_password(length=10):  
 lowercase\_letters = list(range(97, 123))  
 uppercase\_letters = list(range(65, 91))  
 digits = list(range(48, 58))  
 special\_characters = "!#$%&'()\*+,-./:;<=>?@[\]^\_`{|}~"  
  
 password = (  
 chr(random.choice(lowercase\_letters)) +  
 chr(random.choice(uppercase\_letters)) +  
 chr(random.choice(digits)) +  
 random.choice(special\_characters)  
 )  
  
 remaining\_length = length - 4  
 all\_characters = lowercase\_letters + uppercase\_letters + digits + [ord(char) for char in special\_characters]  
 password += ''.join(chr(random.choice(all\_characters)) for \_ in range(remaining\_length))  
  
 password\_list = list(password)  
 random.shuffle(password\_list)  
  
 password = ''.join(password\_list)  
  
 return password  
  
random\_password = generate\_random\_password()  
print("Random Password:", random\_password)



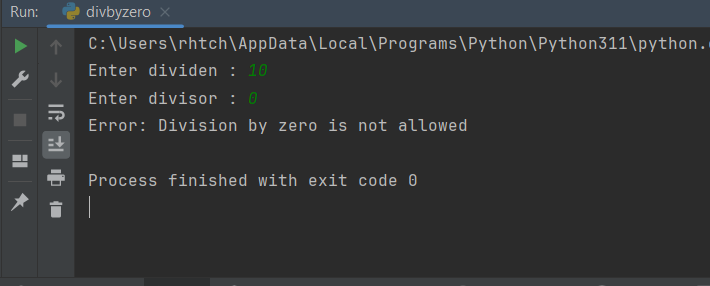
**Write a Python function that takes two numbers and an operator (as a string) and performs the corresponding arithmetic operation (addition, subtraction, multiplication, or division).**

def arithmetic\_operation(num1, num2, operator):  
 if operator == '+':  
 return num1 + num2  
 elif operator == '-':  
 return num1 - num2  
 elif operator == '\*':  
 return num1 \* num2  
 elif operator == '/':  
 if num2 == 0:  
 return "Division by 0 is not allowed"  
 else:  
 return num1 / num2  
 else:  
 return "Unsupported operand"  
  
num1 = int(input("Enter num1 : "))  
num2 = int(input("Enter num2 : "))  
operator = input("Enter operand (+,-,\*,/) : ")  
print(arithmetic\_operation(num1,num2,operator))



**Write a Python function that divides two numbers and handles the case where the divisor is zero by returning a custom error message.**

def divide\_numbers(dividend, divisor):  
 try:  
 result = dividend / divisor  
 except ZeroDivisionError:  
 return "Error: Division by zero is not allowed"  
 else:  
 return result  
  
# Example usage:  
dividend = int(input("Enter dividen : "))  
divisor = int(input("Enter divisor : "))  
print(divide\_numbers(dividend, divisor))



**Write a Python function to compute the nth Fibonacci number using recursion.**

def fibonacci(n):  
 if n <= 0:  
 return 0  
 elif n == 1:  
 return 1  
 else:  
 return fibonacci(n - 1) + fibonacci(n - 2)  
  
n = 5  
print(f"{n}th fibonacci number is {fibonacci(n)}")

