PRACTICE QUESTIONS

1.Write a function that accepts a list of numbers and returns a new list with only even numbers.  
  
def get\_even\_numbers(numbers):  
 return [num for num in numbers if num % 2 == 0]  
  
print( get\_even\_numbers([1, 2, 3, 4, 5, 6]))

2.Create a function to calculate the factorial of a given number without using recursion.  
  
def factorial(n):  
 result = 1  
 for i in range(1, n + 1):  
 result \*= i  
 return result  
  
print( factorial(5))

3.Write a function that takes a sentence as input and returns the number of vowels in it.  
  
def count\_vowels(sentence):  
 vowels = "aeiouAEIOU"  
 return sum(1 for ch in sentence if ch in vowels)  
  
print( count\_vowels("Hello World"))

4.Implement a function that receives a list of strings and returns a dictionary with word lengths as keys and words of that length as values.  
  
def words\_by\_length(words):  
 result = {}  
 for word in words:  
 result.setdefault(len(word), [ ]).append(word)  
 return result  
  
print(words\_by\_length(["hi", "hello", "bye", "yes"]))

5.Define a function that accepts a list of integers and returns the second highest number.  
  
def second\_highest(nums):  
 unique\_nums = list(set(nums))  
 unique\_nums.sort()  
 return unique\_nums[-2] if len(unique\_nums) >= 2 else None  
  
print(second\_highest([10, 20, 4, 45, 99, 99]))

6.Write a program that takes two numbers as input and divides them. Handle the case when the denominator is zero.  
  
def safe\_divide(a, b):  
 try:  
 return a / b  
 except ZeroDivisionError:  
 return "Error: Division by zero not allowed"  
  
print(safe\_divide(10, 2))  
print(safe\_divide(5, 0))

7.Create a calculator using functions for add, subtract, multiply, and divide. Handle invalid inputs.  
  
def add(a, b): return a + b  
def subtract(a, b): return a - b  
def multiply(a, b): return a \* b  
def divide(a, b):  
 try:  
 return a / b  
 except ZeroDivisionError:  
 return "Error: Division by zero"  
  
print(add(5, 3), subtract(5, 3), multiply(5, 3), divide(5, 0))

8.You are reading a file line by line. Write a function that safely reads the file and prints lines, handling errors.  
  
def safe\_read\_file(filename):  
 try:  
 with open(filename, "r") as file:  
 for line in file:  
 print(line.strip())  
 except FileNotFoundError:  
 print("Error: File Not Found")  
 except PermissionError:  
 print("Error: Permission Denied")

9.Write a function to convert a string to an integer. Use try-except to handle invalid inputs.  
  
def str\_to\_int(s):  
 try:  
 return int(s)  
 except ValueError:  
 return "Error: Invalid integer input"  
  
print( str\_to\_int("123"))  
print( str\_to\_int("abc"))

10.Given monthly sales in a list, calculate total annual sales.  
  
def total\_sales(sales):  
 return sum(sales)  
  
print( total\_sales([100, 200, 150, 175, 120, 300]))

11.Find the average score in a list of student scores, excluding empty cells.  
  
def average\_score(scores):  
 valid\_scores = [s for s in scores if s is not None]  
 return sum(valid\_scores) / len(valid\_scores) if valid\_scores else 0  
  
print(average\_score([80, 90, None, 70, 100]))

12.String Task: Extract first name from full names.  
  
def extract\_first\_names(names):  
 return [name.split()[0] for name in names]  
  
print(extract\_first\_names(["Neha Mishra", "Anjana mohan", "Bobby singh"]))

13.Extract domain names from email addresses.  
  
def extract\_domains(emails):  
 return [email.split("@")[1] for email in emails]  
  
print(extract\_domains(["nehasd@gmail.com", "pooja@gmail.com"]))

14.Concatenate first name in Column A and last name in Column B with a hyphen, in uppercase.  
  
def concat\_names(first\_names, last\_names):  
 return [f"{f}-{l}".upper() for f, l in zip(first\_names, last\_names)]  
  
print(concat\_names(["John", "Alice"], ["Doe", "Smith"]))