Ex 2

```
your code here
def digit_sum(number):
    # Convert the number to a string to iterate over its digits
    number_str = str(number)

    # Initialize the sum of digits
    sum_of_digits = 0

    # Iterate over each digit and add it to the sum
    for digit in number_str:
        sum_of_digits += int(digit)

    return sum_of_digits

try:
    # Get user input for a number
    user_number = int(input("Enter a number: "))

    # Calculate the sum of digits
    result = digit_sum(user_number)

    # Print the result
    print("The sum of digits in {} is: {}".format(user_number, result))

except ValueError:
    print("Error: Please enter a valid number.")
except Exception as e:
    print("An error occurred:", str(e))
```

ex3

```
import random

def count_deeper_readings(depth_measurements):
    # Initialize a counter for deeper readings
    deeper_count = 0

    # Iterate over the depth measurements
    for i in range(1, len(depth_measurements)):
        # Check if the current depth is deeper than the previous one
        if depth_measurements[i] > depth_measurements[i - 1]:
            deeper_count += 1

    return deeper_count

try:
    # Generate a random list of depth measurements
    depth_reads = [random.randint(150, 400) for _ in range(500)]

    # Calculate the count of deeper readings
    result = count_deeper_readings(depth_reads)

    # Print the result
    print("Number of readings deeper than the previous one: {}".format(result))

except Exception as e:
    print("An error occurred:", str(e))
```

ex4

```
def fibonacci_sequence(n):
    # Initialize the first two numbers in the Fibonacci sequence
    fibonacci_numbers = [0, 1]

    # Calculate the Fibonacci sequence up to the Nth number
    for i in range(2, n):
        next_number = fibonacci_numbers[-1] + fibonacci_numbers[-2]
        fibonacci_numbers.append(next_number)

    return fibonacci_numbers

try:
    # Get user input for the number of Fibonacci numbers to generate
    N = int(input("Enter the number of Fibonacci numbers to generate: "))

    # Calculate the Fibonacci sequence
    result = fibonacci_sequence(N)

    # Print the result
    print("First {} Fibonacci numbers: {}".format(N, result))

except ValueError:
    print("Error: Please enter a valid integer for the number of Fibonacci numbers.")
except Exception as e:
    print("An error occurred:", str(e))
```

ex5

```
def find_multiples(numbers):
    multiples_dict = {"3": [], "5": [], "7": [], "9": []}

    for num in numbers:
        if num % 3 == 0:
            multiples_dict["3"].append(num)
        if num % 5 == 0:
            multiples_dict["5"].append(num)
        if num % 7 == 0:
            multiples_dict["7"].append(num)
        if num % 9 == 0:
            multiples_dict["9"].append(num)

    return multiples_dict

try:
    # Generate a list of numbers using the provided nums list comprehension
    nums = [n for n in range(50, 150)]

    # Find the multiples and create a dictionary
    result_dict = find_multiples(nums)

    # Convert the dictionary to a string
    result_str = str(result_dict)

    # Save the result to a file called "multiples.txt"
    with open("multiples.txt", "w") as file:
        file.write(result_str)

    print("Result saved to multiples.txt.")

except Exception as e:
    print("An error occurred:", str(e))
```