COSC 1104 – Assignment 1

# I have selected option1 out the two provided options

# Option 1:

Create an application that will analyze whether a number is a prime number or not and provide some additional information around it.

First, you will need to define a function that determines if a number is prime or not. There are many versions of this already; you are encouraged to write this function yourself, but if you use one found on the web please test it thoroughly and cite where it came from.

When the script runs, request input from the user. They should enter a positive whole number. If they enter something that is not a positive whole number, they should be prompted again. There should be no way to cause a crash.

If the user enters a positive whole number, output should include the following:

1. The prime number *before* the number they entered, if there is one.
2. Whether their number is prime.  
   If it is not prime, display a list of the number’s divisors.
3. The next prime number *after the* number they entered.

Sample output:

Please enter the number to check: kyle

That is not a positive whole number. Try again.

Please enter the number to check: 7

The prime number before 7 is 5.

7 is a prime number.

The prime number after 7 is 11.

Press Enter to exit the program...

Please enter the number to check: 12

The prime number before 12 is 11.

12 is not prime. Its factors are 2, 3, 4, 6.

The prime number after 12 is 13.

Press Enter to exit the program...

def is\_prime(number): """is\_prime function iteratively checks for factors"""

    """Check if a number is prime using iteration."""

    if number <= 1:

        return False

    for i in range(2, number):

        if number % i == 0:

            return False

    return True

def find\_previous\_prime(number):"""find\_previous\_prime use loops to find nearby primes"""

    """Find the largest prime number less than the given number."""

    for n in range(number - 1, 1, -1):

        if is\_prime(n):

            return n

    return None

def find\_next\_prime(number): """find\_next\_prime use loops to find primes if nearby"""

    """Find the smallest prime number greater than the given number."""

    n = number + 1

    while True:

        if is\_prime(n):

            return n

        n += 1

def get\_divisors(number): """get\_divisors function results a list of divisors"""

    """Get a list of divisors of the number."""

    divisors = []

    for i in range(1, number + 1):

        if number % i == 0:

            divisors.append(i)

    return divisors

def main():

    while True:

        user\_input = input("Enter a positive whole number: ").strip()

        if not user\_input.isdigit():

            print("Invalid input. Please enter a positive whole number.")

            continue

        number = int(user\_input)

        if number <= 0:

            print("Invalid input. Please enter a positive whole number.")

            continue

        previous\_prime = find\_previous\_prime(number)

        next\_prime = find\_next\_prime(number)

        prime\_status = is\_prime(number)

        print(f"\nResults for number {number}:")

        if previous\_prime:

            print(f"a) The prime number before {number} is {previous\_prime}.")

        else:

            print(f"a) There is no prime number before {number}.")

        if prime\_status:

            print(f"b) {number} is a prime number.")

        else:

            divisors = get\_divisors(number)

            print(f"b) {number} is not a prime number.")

            print(f"   Divisors of {number}: {divisors}")

        print(f"c) The next prime number after {number} is {next\_prime}.")

        break

if \_\_name\_\_ == "\_\_main\_\_":

    main()