ASSIGNMENT 10.2

NAME:ABHINAND REDDY

ROLLNO:2403A51288

BATCH:12 BATCH

Task Description#1 AI-Assisted Code Review (Basic Errors)

• Write Python program as shown below.

• Use an AI assistant to review and suggest corrections

```
def calcFact(n):

result=1

x=0

for i in range(1,n):

result=result i

return result

def main():

num = 5

FACT = calcFact (num)
```

print("the factorial of", num, "is", FACT)

t=10

if FACT>10:

print("BIG Number")

else:
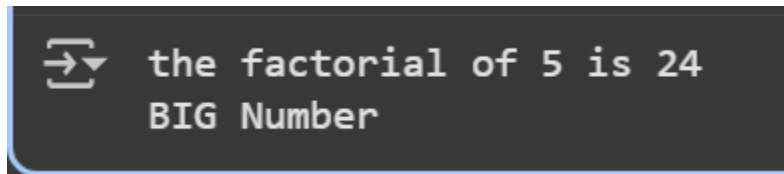
print("small number")

main()

CODE:

```python
def calcFact(n):
    result=1
    x=0
    for i in range(1,n):
        result=result*i
    return result

def main():
    num = 5
    FACT = calcFact(num)
    print("the factorial of",num,"is",FACT)
    t=10
    if FACT>10:
        print("BIG Number")
    else:
        print("small number")

main()
```

OUTPUT:

```
→▼  the factorial of 5 is 24
    BIG Number
```

Task Description#2 Automatic Inline Comments

• Write the Python code for Fibonacci as shown below and execute.

• Ask AI to improve variable names, add comments, and apply PEP8 formatting

(cleaned up).

• Students evaluate which suggestions improve readability most.

one.

```
def f1(xx):

b-1

c=2

Zz-[a,b]

while c<-XX:

death

Zz.append(d)

b-d

return zz

def m():
```

NN-10

ans-f1(NN)

print("fib series till", NN, ":", ans)

m()

code:

```python
def f1(xx):
    b = 1
    c = 2
    Zz = [0, b]
    while c < xx:
        d = b + Zz[c-2]
        Zz.append(d)
        b = d
        c += 1
    return Zz

def m():
    NN = 10
    ans = f1(NN)
    print("fib series till", NN, ":", ans)

m()
```

Output:

```
fib series till 10 : [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

Task Description#3

Prompt: Create a Python script (add, subtract, multiply, divide) with NumPy-style docstrings, compare with AI docstrings, and explain code smells.

Code:

```python
"""
AI-Generated Module Docstring
-----------------------------
This module provides a simple calculator with basic arithmetic functions:
addition, subtraction, multiplication, and division. Each function includes
manual NumPy-style docstrings for documentation practice.

Students can compare these manual docstrings with AI-generated ones to
understand structured documentation techniques.
"""

def add(a, b):
    """
    Add two numbers.

    Parameters
    ----------
    a : float
        First number.
    b : float
        Second number.

    Returns
    -------
    float
        The sum of a and b.
    """
    return a + b
```

```python
def subtract(a, b):
    """
    Subtract one number from another.

    Parameters
    ----------
    a : float
        Number from which to subtract.
    b : float
        Number to subtract.

    Returns
    -------
    float
        The difference of a and b.
    """
    return a - b
```

```python
def multiply(a, b):
    """
    Multiply two numbers.

    Parameters
    ----------
    a : float
        First number.
    b : float
        Second number.

    Returns
    -------
    float
        The product of a and b.
    """
    return a * b
```

```python
def divide(a, b):
    """
    Divide one number by another.

    Parameters
    ----------
    a : float
        Numerator.
    b : float
        Denominator (must not be zero).

    Returns
    -------
    float
        The result of division.

    Raises
    ------
    ValueError
        If b is zero.
    """
    if b == 0:
        raise ValueError("Division by zero is not allowed.")
    return a / b


if __name__ == "__main__":
    # Example usage
    x, y = 10, 5
    print("Add:", add(x, y))
    print("Subtract:", subtract(x, y))
    print("Multiply:", multiply(x, y))
    print("Divide:", divide(x, y))
```

Output:

```
Add: 15
Subtract: 5
Multiply: 50
Divide: 2.0
```