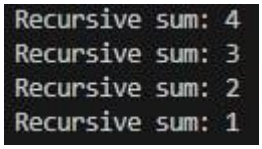
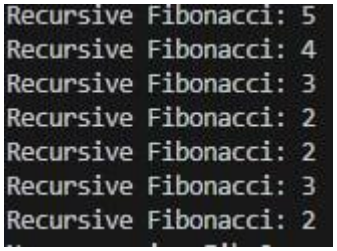
Activity No. <n>	
<Replace with Title>	
Course Code: CPE010	**Program:** Computer Engineering
Course Title: Data Structures and Algorithms	**Date Performed**: August 12, 2025
Section: CPE21S4	**Date Submitted:**
Name(s): Punay, Heidee S.	**Instructor:** Engr. Jimlord Quejado

6. Output

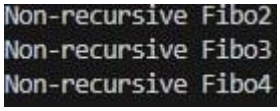
Recursive Sum:



Recursive Fibo:



Non-Recursive Fibo:



7. Supplementary Activity

idee > Data struc n algo > recursive.cpp > nonrecFunc(int)

```cpp
#include<iostream>

int recSum(int num);
int recFibo(int num);
int nonrecFunc(int num);

int main()
{
    recSum(4);
    recFibo(5);
    nonrecFunc(4);

    return 0;
}

int recSum(int num)
{
    if(num<=0)
    {
        return 0;
    }
    else
    {
    std::cout<<"Recursive sum: "<<num<<std::endl;
    return num + recSum(num-1);
    }
}

int recFibo(int num)
{
    if(num<=1)
    {
        return num;
    }
    else{
        std::cout<<"Recursive Fibonacci: "<<num<<std::endl;
        return recFibo(num - 1) + recFibo(num-2);
```

```
    else{
        std::cout<<"Recursive Fibonacci: "<<num<<std::endl;
        return recFibo(num - 1) + recFibo(num-2);
    }
}

int nonrecFunc(int num)
{
    if(num<=1)
    {
        return num;
    }
    int x = 0, y =1, z;
    for(int i = 2; i <=num; i++)
    {
        z = x + y * i;
        std::cout<<"Non-recursive Fibo"<<z<<std::endl;
    }
    return y;
}
```

8. Conclusion

To conclude, in recursion we can make the codes into smaller version. And learned that there is another function aside from using for loop. In this code it only uses one loop so its an O(n).

9. Assessment Rubric