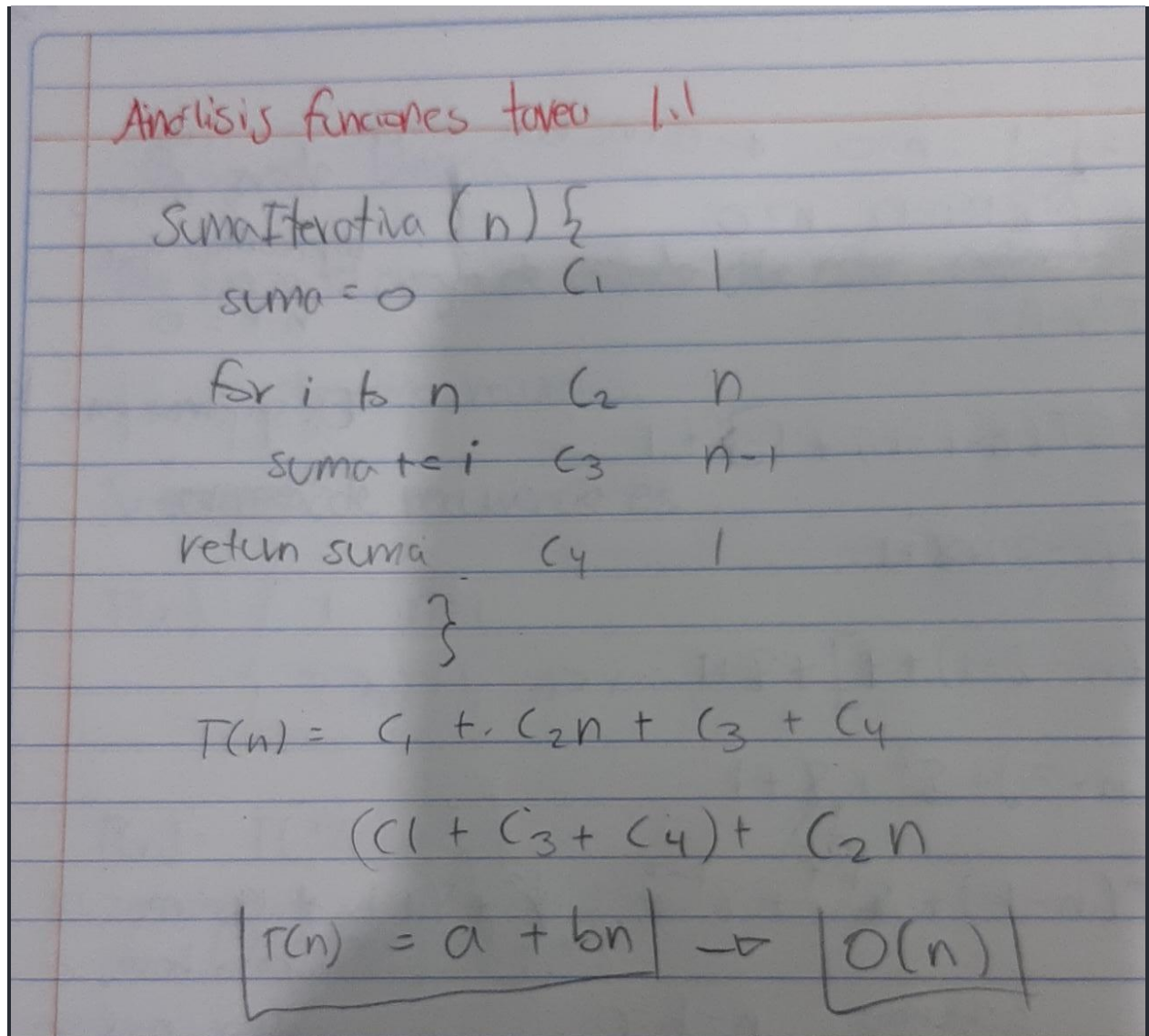


- Objetivo: Sumar de manera iterativa de 1 hasta 'n'.
- {Param} int n - número hasta el cual se quiere sumar
- Complejidad computacional: $O(n)$ LINEAL

```
long long sumaIterativa(long long n)
{
    long long suma = 0; // 1

    for (int i = 1; i <= n; i++) // n
    {
        suma = suma + i; // n - 1
    }
    return suma; // 1
}
```

Análisis:



- Objetivo: Sumar de manera recursiva todos los números del 1 hasta 'n'.
- {Param} int n - número hasta el cual se quiere sumar.
- Retorna: int suma - suma total de todos los números de 1 hasta 'n'.
- Complejidad computacional: $O(n)$ LINEAL

```
long long sumaRecursiva(long long n)  
{  
    // Caso base  
    if (n == 0 || n == 1) // 1
```

```

{
    return 1; // 1
}
// Condición recursiva
else
{
    // La operación que queremos que haga la función se declara en el estatuto de retorno.
    return n + sumaRecursiva(n - 1); // n veces se repite.
}
}

```

Análisis:

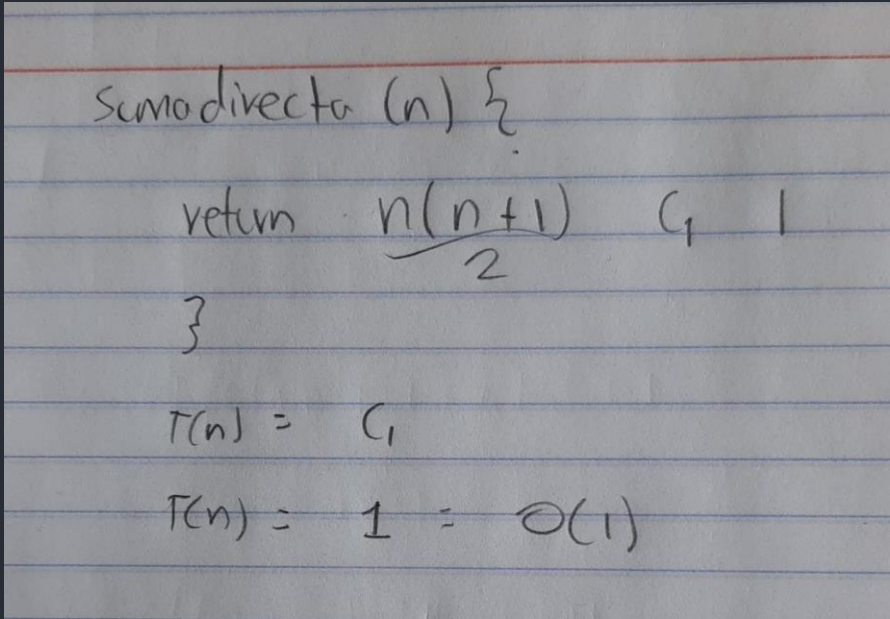
$SumaRecursiva(n) \{$
 if $n == 0$ or $n == 1$ C_1 1
 return 1 C_2 1
 else
 return $n + SumaRecursiva(n-1)$ C_3 n
 $T(n) = (C_1 + C_2) + C_3 n$
 $T(n) = a + bn \rightarrow O(n)$
 $n - k = 1$
 $k = n - 1$
 $T(n) = \begin{cases} 1, & n=1 \\ T(n-1) + 1, & n > 1 \end{cases}$
 $T(n-1) + 1 + 1$
 $T(n-2) + 2$
 $T(n-3) + 3$
 $T(n-k) + k$
 $T(n-(n-1)) + n-1$
 $T(1) + n-1$
 $1 + n-1$
 $O(n)$

- Objetivo: Sumar de manera recursiva todos los números del 1 hasta 'n'.

- {Param} int n - número hasta el cual se quiere sumar.
- Complejidad computacional: $O(1)$ constante

```
long long sumaDirecta(long long n)
{
    return (n * (n + 1)) / 2; //  $O(1)$ 
}
```

Análisis:

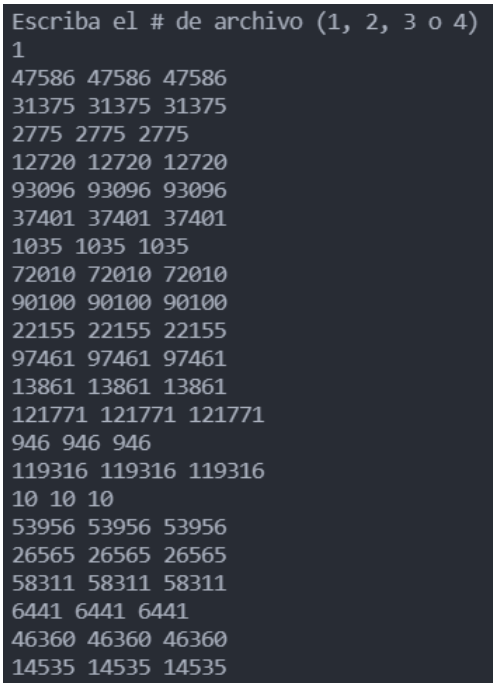


Handwritten analysis of the `sumaDirecta` function:

$$\text{sumaDirecta}(n) \{$$
$$\quad \text{return } \frac{n(n+1)}{2} \quad C_1 \quad |$$
$$\quad \}$$
$$T(n) = C_1$$
$$T(n) = 1 = O(1)$$

OUTPUT DEL PROGRAMA

Caso de prueba 1:



Escriba el # de archivo (1, 2, 3 o 4)

1

47586 47586 47586

31375 31375 31375

2775 2775 2775

12720 12720 12720

93096 93096 93096

37401 37401 37401

1035 1035 1035

72010 72010 72010

90100 90100 90100

22155 22155 22155

97461 97461 97461

13861 13861 13861

121771 121771 121771

946 946 946

119316 119316 119316

10 10 10

53956 53956 53956

26565 26565 26565

58311 58311 58311

6441 6441 6441

46360 46360 46360

14535 14535 14535

Caso de prueba 2:

```

Escriba el # de archivo (1, 2, 3 o 4)
2
854778 854778 854778
780625 780625 780625
576201 576201 576201
671061 671061 671061
1023165 1023165 1023165
809628 809628 809628
545490 545490 545490
950131 950131 950131
1013176 1013176 1013176
731445 731445 731445
1037520 1037520 1037520
679195 679195 679195
1113778 1113778 1113778
543403 543403 543403
1106328 1106328 1106328
503506 503506 503506
881128 881128 881128
755835 755835 755835
898470 898470 898470
618828 618828 618828
849556 849556 849556
683865 683865 683865

```

Caso de prueba 3:

```

Escriba el # de archivo (1, 2, 3 o 4)
3
141246028 141246028 141246028
116273625 116273625 116273625
50737701 50737701 50737701
93277311 93277311 93277311
179181915 179181915 179181915
63534628 63534628 63534628
153904740 153904740 153904740
59170881 59170881 59170881
160625926 160625926 160625926
156813195 156813195 156813195
104264020 104264020 104264020
164992695 164992695 164992695
105016278 105016278 105016278
85053403 85053403 85053403
161775078 161775078 161775078
78168756 78168756 78168756
75983628 75983628 75983628
68790585 68790585 68790585
177482220 177482220 177482220
79537578 79537578 79537578
102295056 102295056 102295056
86717865 86717865 86717865
156813195 156813195 156813195
147189903 147189903 147189903

```

Jorge Del Barco Garza A01284234

Caso de prueba 4:

```
Escriba el # de archivo (1, 2, 3 o 4)
4
141262836 141262836 141262836
```

Por motivos de recursos el programa termina en la primera iteración.