

Ejercicios de Funciones

C++

Serie Fibonacci

La serie Fibonacci $a(1), a(2), \dots, a(n), \dots$ está definida por:

- $a(1) = 1$
- $a(2) = 1$
- $a(n) = a(n-1) + a(n-2)$ para todo $n > 2$

Escribir una función `mostrar_serie_fibonacci` que genere la serie, la función tendrá como parámetro un número entero positivo n , que corresponderá a la cantidad de números de la serie que se quiere generar, si por ejemplo la llamada a la función es:

```
mostrar_serie_fibonacci(4);
```

Entonces se obtendrá:

```
1, 1, 2, 3
```

Números Narcisistas 1

Sea un número ***n*** de ***k*** dígitos, por ejemplo el número **324** que tiene **3** dígitos, se dice que es narcisista si:

$$n = d_k^k + d_{k-1}^k + \dots + d_1^k$$

Ejemplo: **153** es un número de **3** dígitos, por lo tanto **n = 153** y **k = 3**

$$153 = 1^3 + 5^3 + 3^3$$

Escribir una función **es_narcisista**, que tenga un parámetro para recibir el número y que devuelva **true** o **false** verificando si el número es narcisista.

Números Narcisistas 2

Usando la función escrita en el ejercicio anterior, Escribir una función `mostrar_narcisistas` que tenga 2 parámetros que definen un **límite inferior** y un **límite superior** y que devuelva todos los números narcisistas que se encuentre en ese rango, incluido los límites.

Ejemplo:

```
mostrar_narcisistas(1, 1000);
```

La función retornara:

```
1, 2, 3, 4, 5, 6, 7, 8, 9, 153, 370, 371, 407
```

Siguiente Primo

Escribir una función `siguiente_primo`, que tenga como parámetro un número y que devuelva el siguiente número primo, si el número ingresado es primo deberá devolver el siguiente sin considerar el número dado como parámetro.

Ejemplo:

```
cout << siguiente_primo(10) << endl; // debe imprimir 11  
cout << siguiente_primo(11) << endl; // debe imprimir 13
```