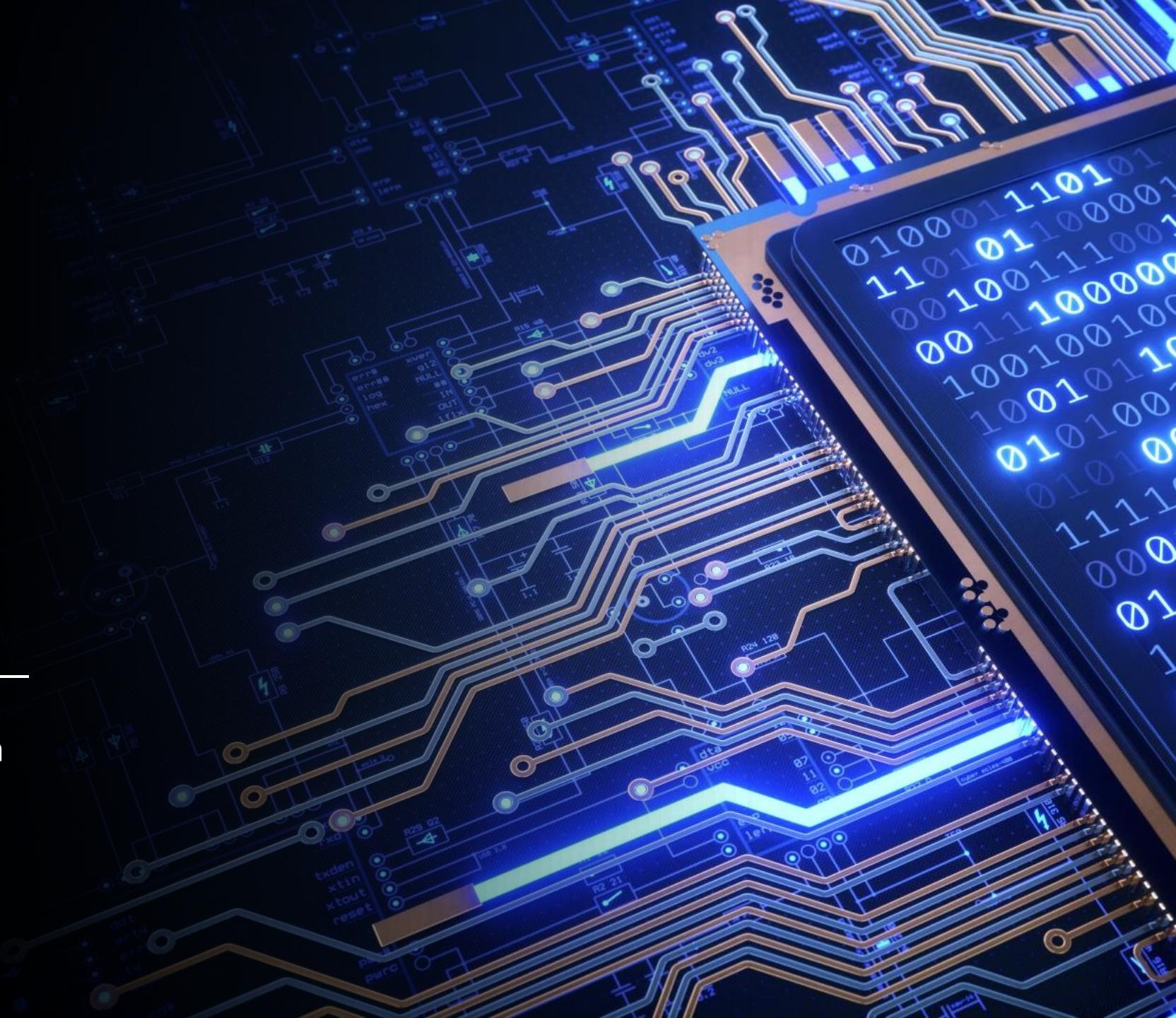


Ejercicios con Estructuras de Control Condicionales e Iterativas

EIF200 Fundamentos de Informática
2-2024 / 21-08-2024
G.04 G.05



Desafío 11.1

Función **diasRestriccion()**

| Restricción vehicular sanitaria del 4 al 31 de enero 2021 | | | | | | | | | | |
|--|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 |
| Lunes | x | x | . | . | . | . | . | . | . | . |
| Martes | . | . | x | x | . | . | . | . | . | . |
| Miércoles | . | . | . | . | x | x | . | . | . | . |
| Jueves | . | . | . | . | . | . | x | x | . | . |
| Viernes | . | . | . | . | . | . | . | . | x | x |
| Sábado | x | . | x | . | x | . | x | . | x | . |
| Domingo | . | x | . | x | . | x | . | x | . | x |

Restricción aplica para todo el país.
De lunes a viernes restricción total de 10 p.m. a 5 a.m.
Fines de semana restricción total de 9 p.m. a 5 a.m.



Escribir una función **diasRestriccion()** en C++ que reciba como parámetro un número entero que corresponde con el número de placa de un vehículo y devuelva dos nombres con los días de la semana en los que el vehículo tiene restricción de circulación, de acuerdo con la tabla que se muestra.

En el main(), invoque la función **diasRestriccion()** y muestre los días de restricción correspondientes a un vehículo, cuyo número de placa se solicita al usuario del programa.

Desafío 17.1

Cantidad de Divisores

Escribir una función en el Lenguaje C++ que reciba como parámetro un número entero **n**, y devuelva la **cantidad de divisores** que tiene.

Desafío 18.0

Es primo

Crear un diagrama de flujo del algoritmo para determinar si un número que se recibe es **primo**.

Luego, escribir una **función** en el Lenguaje C++ que reciba como parámetro un número entero y devuelva true si el número es primo o false si no.

2₃5₇₁₁

Desafío 20

Cuadrado de asteriscos

Escribir un programa en el Lenguaje C++ que lea del usuario un número entero n y despliegue en la pantalla un cuadrado de asteriscos de tamaño n (n asteriscos de lado). Por ejemplo, en la imagen abajo se muestra un cuadrado de tamaño 5.

```
*****  
*****  
*****  
*****  
*****
```



**PROGRAMA
QUE IMPRIMA
CUADRADO DE
ASTERISCOS**

Desafío 20.1

Perímetro del Cuadrado con asteriscos

Escribir una función en el Lenguaje C++ que reciba como parámetro un número entero n y devuelva una hilera o string que permita desplegar en la pantalla, con asteriscos, el perímetro de un cuadrado de tamaño n .

En el `main()`, lea del usuario un número entero n y utilice la función definida para mostrar en la pantalla el perímetro de un cuadrado de tamaño n .

C:\Program Files (x86)\Zinjal\bin\runner.exe

```
5
* * * * *
*      *
*      *
*      *
* * * * *

<< Program finished: exit code: 0 >>
<< Press enter to close this window >>
```

Desafío 21

Triángulo rectángulo de asteriscos

Escribir un programa en el Lenguaje C++ que lea del usuario un número entero n y despliegue en la pantalla un triángulo rectángulo de asteriscos con catetos de tamaño n (catetos de n asteriscos). Por ejemplo, abajo se muestra un triángulo rectángulo con catetos de tamaño 10.

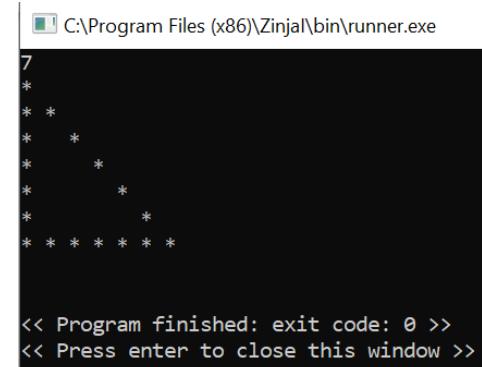


Desafío 21.1

Triángulo rectángulo de asteriscos

Escribir una función en el Lenguaje C++ que reciba como parámetro un número entero n y devuelva una hilera o string que permita desplegar en la pantalla, con asteriscos, el perímetro de un triángulo rectángulo de catetos de tamaño n.

En el main(), lea del usuario un número entero n y utilice la función definida para mostrar en la pantalla el perímetro de un triángulo rectángulo de catetos de tamaño n.

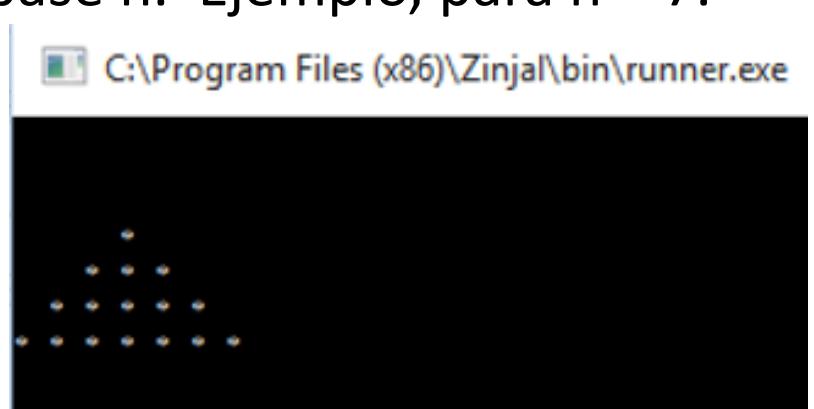


Desafío 21.2

Triángulo Isósceles con asteriscos

Escribir una función en el Lenguaje C++ que reciba como parámetro un número entero n y devuelva una hilera o string que permita desplegar en la pantalla, con asteriscos, un triángulo isósceles de base n .

En el main(), lea del usuario un número entero n y utilice la función definida para mostrar en la pantalla un triángulo isósceles de base n . Ejemplo, para $n = 7$.



El Extraño Reino del Revés

Cuenta la leyenda que en el extraño Reino del Revés los números se leen al revés..., de derecha a izquierda!!!

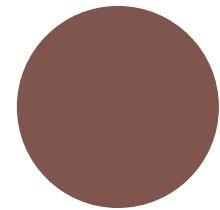
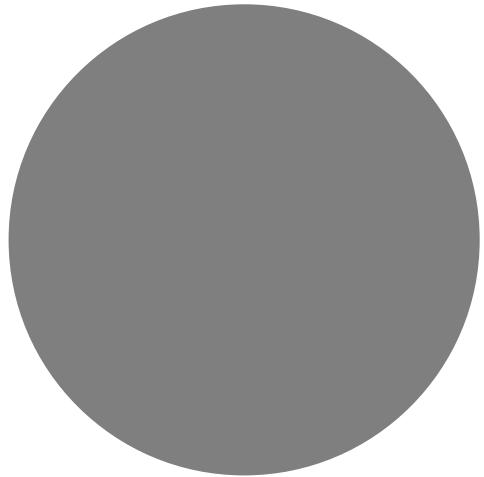
Por ejemplo, lo que para nosotros es el número 2464, en el Reino del Revés es el 4642 y viceversa.

Escriba una función en C++ que nos traduzca los números de como se leen en el Reino del Revés a como los leemos en Nuestro Reino y, viceversa. La función recibe un número entero **n** como parámetro y devuelve un número entero con los dígitos de **n** en el orden inverso.



Desafío 24

El Extraño Reino del
Revés:
función int invertir().

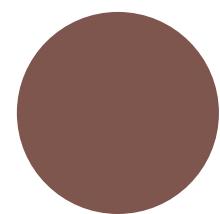
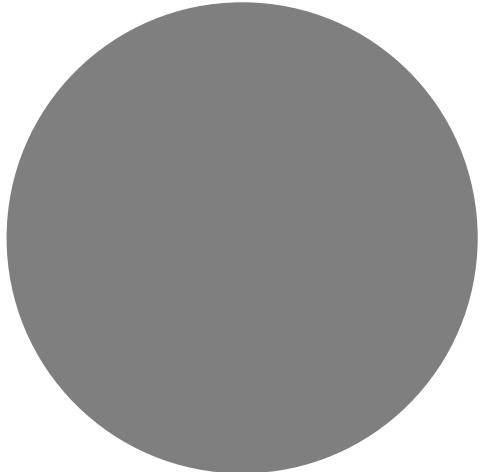


2, 3, 5, 7, 11, 101, 131,
151, 181, 191, 313, 353,
373, 383, 727, 757, 787,
797, 919, 929, 10301,
10501, 10601, 11311,
11411, 12421, 12721...

Números Capicúa

Un **Número** es **Capicúa** si se lee igual de izquierda a derecha que de derecha a izquierda. El término procede de la expresión catalana cap i cua (que significa 'cabeza y cola').

Escriba una función en C++ que reciba un número entero y determine si el número es Capicúa o no.



Desafío 24.1

Es un Número
Capicúa???



Desafío 24.2

incrementaDigitos()

Escriba una función en C++ que reciba como parámetro una variable entera positiva y la devuelva con cada uno de los dígitos del número que contiene incrementados en 1.

Por ejemplo, si recibe una variable con el número 123, devuelve esa misma variable con el número 234.

El incremento de los dígitos es circular. Por ejemplo, el dígito 8 se incrementa a 9 y el dígito 9 se incrementa a 0. Por lo tanto, si la función recibe una variable con el número 9876, devuelve esa misma variable con el número 987.

```
C:\Program Files (x86)\Zinjal\ n = 9876, con dígitos incrementados: 987 << Program finished: exit code: 0 >> << Press enter to close this window >>
```

Desafío 24.4

Ábaco

Escriba una función en C++ que reciba un número entero y devuelva una hilera que permita mostrar la forma como se representaría ese número en un ábaco, utilizando para cada cuenta o bolita del ábaco el símbolo de la letra O mayúscula. Pruebe la función desde el main(). Por ejemplo: para el número 76245 se tendría el siguiente resultado:



```
Número: 76245  
Representación del número 76245 en un ábaco:  
00000  
0000  
00  
000000  
0000000  
  
<< Program finished: exit code: 0 >>  
<< Press enter to close this window >>
```

Desafío 24.4

Ábaco

Número: 76245

Representación del número 76245 en un ábaco:

```
00000  
0000  
00  
000000  
0000000
```

```
<< Program finished: exit code: 0 >>  
<< Press enter to close this window >>
```

Escriba una función en C++ que reciba un número entero y devuelva una hilera que permita mostrar la forma como se representaría ese número en un ábaco, utilizando para cada cuenta o bolita del ábaco el símbolo de la letra O mayúscula. Pruebe la función desde el main(). Por ejemplo: para el número 76245 se tendría el siguiente resultado:

Desafío 25.1

Suma de Divisores Propios

Escribir una función en el Lenguaje C++ que reciba como parámetro un número entero **n**, y devuelva la **suma de sus divisores propios** (distintos de n).

Número perfecto

Un número se dice perfecto si la **suma** de sus **divisores propios** (divisores distintos de él) es **igual** al **número**.

| Pos. | p | Número perfecto | Nº dígitos | Año | Descubridor |
|------|------------|---------------------------|----------------|---|-------------|
| 1 | 2 | 6 | 8 | 1456 | Anónimo |
| 2 | 3 | 28 | 28 | 17 | Cataldi |
| 3 | 5 | 496 | 31 | 1588 | Cataldi |
| 4 | 7 | 8 128 | 47 | 1588 | Cataldi |
| 5 | 13 | 33 050 336 | 8 1456 | anónimo | |
| 6 | 17 | 8 589 669 056 | 10 1588 | Cataldi | |
| 7 | 19 | 137 438 691 288 | 12 1588 | Cataldi | |
| 8 | 31 | 2 388 561 848 437 580 896 | 19 1772 | Alhacén | |
| 9 | 61 | 2658459589...953826096 | 37 1773 | Petrushin | |
| 10 | 89 | 191516942...548166216 | 54 1911 | Powers | |
| 11 | 107 | 131640364...783728128 | 65 1914 | Powers | |
| 12 | 127 | 144740111...19951218 | 77 1978 | Luzac | |
| 13 | 521 | 2 272322007...5376946076 | 314 1982 | Robinson | |
| 14 | 607 | 145451351...5376946076 | 314 1982 | Robinson | |
| 15 | 1270 | 64526282...98429128 | 770 1982 | Robinson | |
| 16 | 2 203 | 108252685...45738272 | 1327 1982 | Robinson | |
| 17 | 2 281 | 994870543...139815776 | 1373 1982 | Robinson | |
| 18 | 3 217 | 335705231...285625056 | 1837 1987 | Riesel | |
| 19 | 4 239 | 102232221...852449256 | 2092 1992 | Riesel | |
| 20 | 4 423 | 407872717...91326498 | 2863 1991 | Huntley | |
| 21 | 9 699 | 114347317...297267716 | 5634 1993 | Gillies | |
| 22 | 9 941 | 598858496...703496576 | 5983 1993 | Gillies | |
| 23 | 11 233 | 359613121...691980538 | 6751 1993 | Gillies | |
| 24 | 19 937 | 831144509...271942604 | 12051 1971 | Tuckerman | |
| 25 | 21 111 | 102533377...102533377 | 12051 1971 | Tuckerman y Nickel | |
| 26 | 23 209 | 111537765...046615866 | 13973 1973 | Tuckerman | |
| 27 | 44 497 | 365093519...031827456 | 26779 1979 | Neimann y Sławiński | |
| 28 | 86 243 | 144145862...306405280 | 5192 1982 | Sloanevi | |
| 29 | 104 961 | 124542628...926482528 | 66330 1987 | Cochell y Welsh | |
| 30 | 134 969 | 124152056...774620560 | 72000 1988 | Sloanevi | |
| 31 | 216 091 | 778373459...848007216 | 13100 1990 | Sloanevi | |
| 32 | 756 839 | 151616370...563731238 | 45663 1992 | Sloanevi y Gage | |
| 33 | 859 433 | 38848822...416793608 | 51740 1994 | Sloanevi y Gage | |
| 34 | 128 797 | 849732889...17870428 | 57587 1996 | Sloanevi y Gage | |
| 35 | 398 589 | 102533377...102533377 | 841942 1996 | Amesbury, Woltman, et al. (GMPS) | |
| 36 | 2008 231 | 194776428...075761808 | 104826 1996 | Amesbury, Woltman, et al. (GMPS) | |
| 37 | 3021 377 | 1189168468...022575686 | 181950 1996 | Clarkson, Woltman, Kurwasvi, et al. (GMPS) | |
| 38 | 6972 953 | 955701360...123572736 | 4187919 1997 | Baumhoff, Woltman, Kurwasvi, et al. (GMPS) | |
| 39 | 13 486 917 | 274761459...020561876 | 2011 2000 | Cameron, Woltman, Kurwasvi, et al. (GMPS) | |
| 40 | 20 998 616 | 442028867...926981268 | 12028 2003 | Stader, Woltman, Kurwasvi, et al. (GMPS) | |
| 41 | 24 954 651 | 442028867...926981268 | 12028 2003 | Stader, Woltman, Kurwasvi, et al. (GMPS) | |
| 42 | 25 964 951 | 746209461...791588128 | 1593208 2005 | Novak, Woltman, Kurwasvi, et al. (GMPS) | |
| 43 | 30 402 497 | 442749374...164704256 | 1830403 2005 | Cooper, Boone, Woltman, Kurwasvi, et al. (GMPS) | |
| 44 | 32 562 567 | 775048068...207511696 | 207511696 2007 | Cooper, Boone, Woltman, Kurwasvi, et al. (GMPS) | |

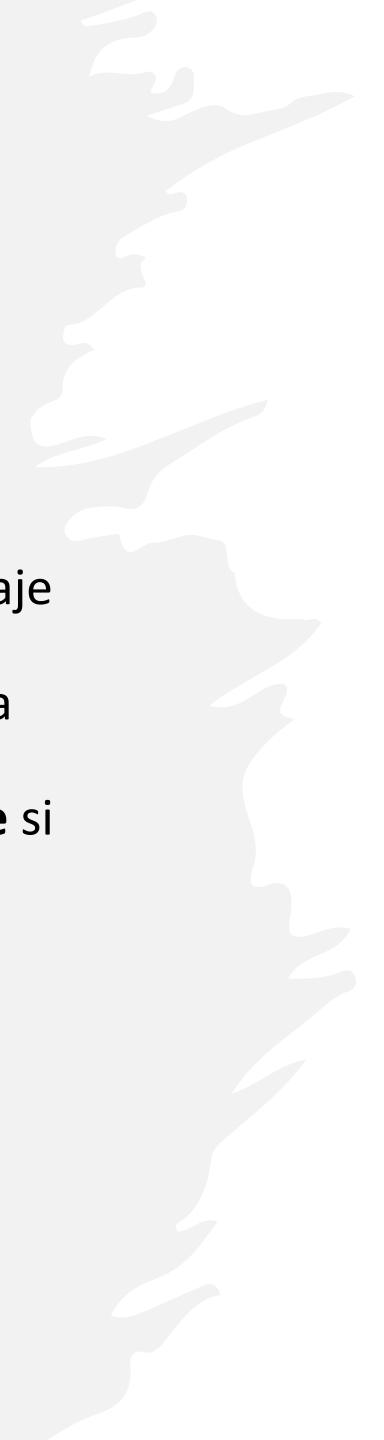


Desafío 25.2 numeroPerfecto()

Escribir una función en el Lenguaje C++ que reciba como parámetro un número entero **n**, y devuelva **true** si el número es perfecto o **false**, si no.

Desafío 26.3 palindromo()

Escribir una función en el Lenguaje C++ que reciba como parámetro una hilera de texto que contenga una palabra y devuelva **true** si la palabra es un palíndromo o **false** si no.



RADAR
OSO ALA
RECONOCER



Ejercicios con Funciones y Estructuras de Control Iterativas

EIF200 Fundamentos de Informática
2-2024 / 21-08-2024
G.04 G.05

