

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										
	Terminación de placa									
	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.

mopt »

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.



2 3 5 7 11

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.



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 .

```
C:\Program Files (x86)\Zinjal\bin\r\nrunner.exe
```

```
7  
*  
* *  
*   *  
*     *  
*       *  
*         *  
*           *  
*             *  
*               *  
*                 *  
*                   *  
*                     *  
*                       *  
*                         *  
*                           *  
*                             *
```

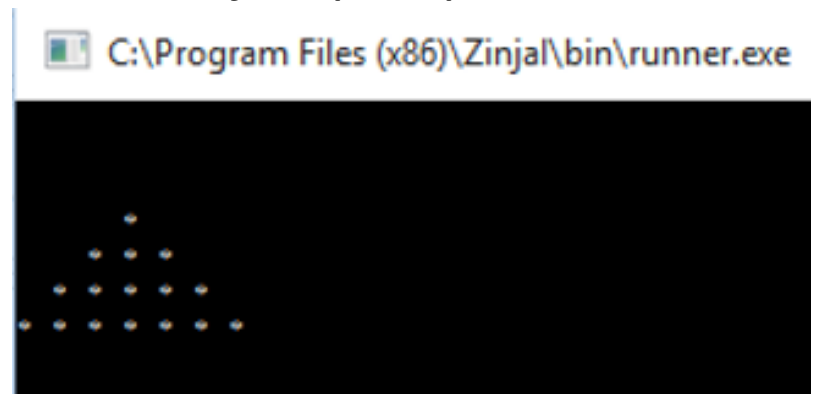
```
<< Program finished: exit code: 0 >>  
<< Press enter to close this window >>
```


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 $n = 7$.



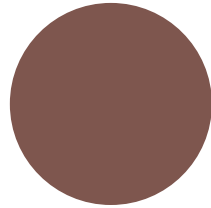
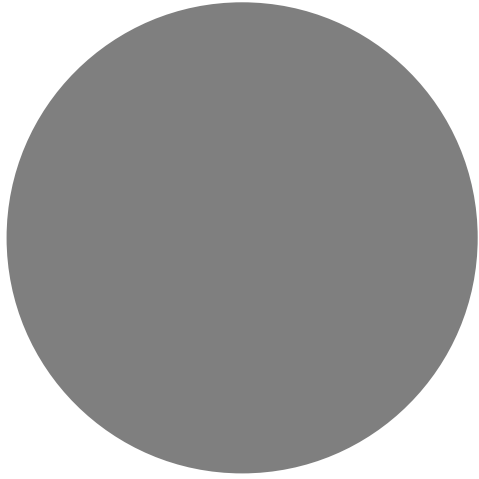


El Extraño Reino del Revés

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

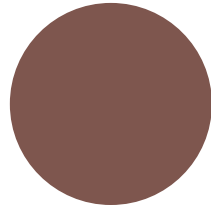
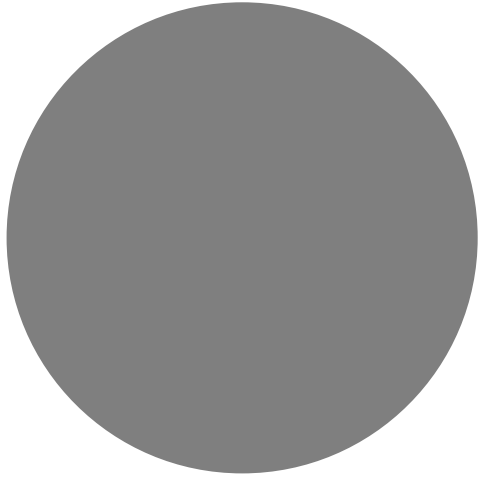
El Extraño Reino del
Revés:
función `int invertir()`.

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???

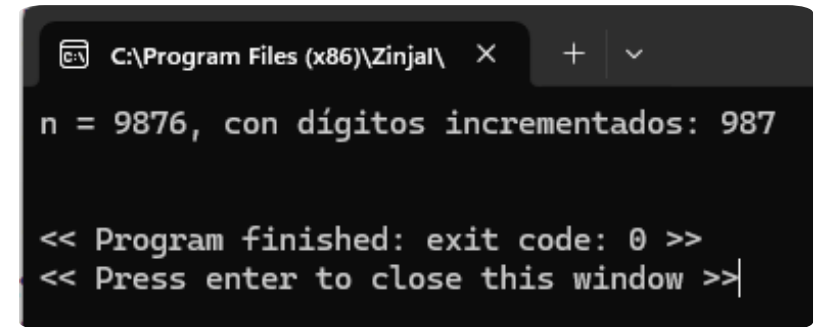
Desafío 24.2

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.

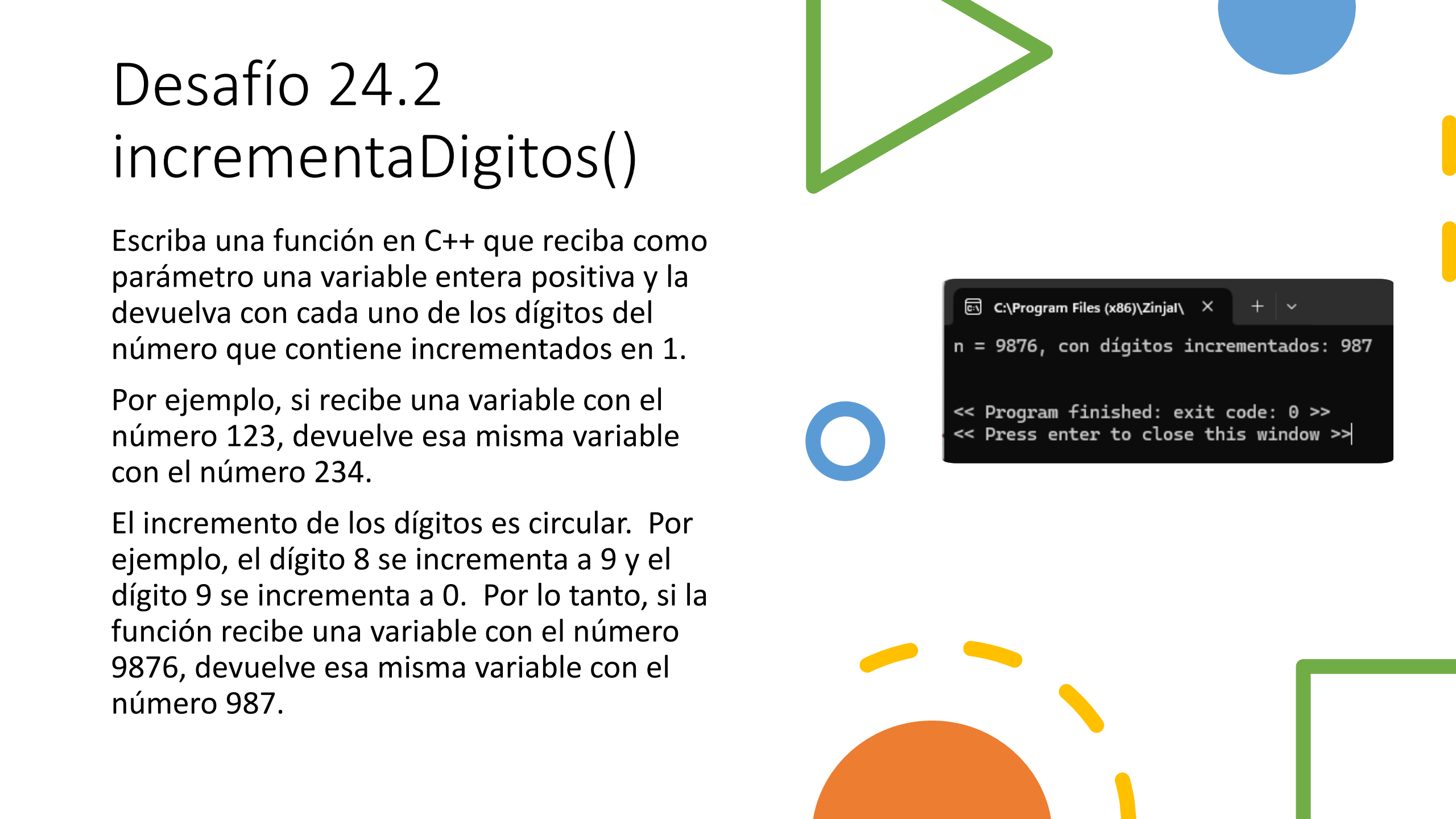
Por ejemplo, si recibe una variable con el número 123, devuelve esa misma variable con el número 234.

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.

A terminal window with a dark background and light-colored text. The title bar shows the file path 'C:\Program Files (x86)\Zinjal\' and standard window controls. The output text is as follows:

```
n = 9876, con dígitos incrementados: 987

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

The background of the slide features several abstract geometric shapes: a large green triangle pointing right in the top left, a blue circle in the top right, a blue circle in the middle left, a large orange circle in the bottom left, and a green L-shaped line in the bottom right. There are also several yellow dashed lines scattered across the bottom right area.

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).

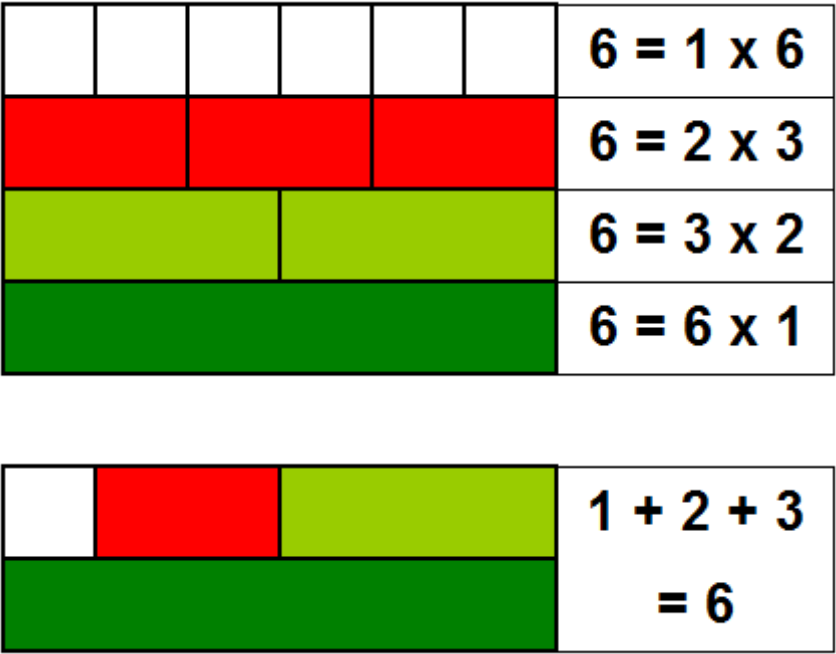
Número perfecto

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

Lista de números perfectos www.vaxasoftware.com

Los números perfectos son generados por la formula $2^{p-1}(2^p - 1)$, donde p y $2^p - 1$ son primos.

Pes.	p	Número perfecto	Nº dígitos	Año	Descubridor
1	2	6	1	?	
2	3	28	2	?	
3	5	496	3	?	
4	7	8 128	4	?	
5	13	33 550 336	8	1456	arabísimo
6	17	8 589 869 056	10	1588	Cataldi
7	19	137 438 691 328	12	1588	Cataldi
8	31	2 305 843 208 139 952 128	19	1772	Euler
9	61	268445990 953842176	37	1863	Pervushin
10	89	19 156 1842 548 169 216	54	1811	Powers
11	107	13 164 0364 763 728 128	60	1914	Powers
12	127	144 740 111 199 152 128	77	1876	Lucas
13	521	235627234 555646976	314	1852	Robinson
14	607	14 105 3783 537 328 128	366	1952	Robinson
15	1 279	54 162 362 984 291 328	770	1862	Robinson
16	2 203	108925835 453 782 528	1327	1952	Robinson
17	2 281	894970543 139 157 776	1373	1952	Robinson
18	5 217	335 763 21 628 525 056	1937	1867	Reese
19	4 253	18 201 7480 133 375 336	2561	1961	Hurwitz
20	4 423	40 787 2717 91 253 4528	2663	1961	Hurwitz
21	9 869	11 434 7317 429 572 716	9834	1963	Gilles
22	9 941	98888496 073 496 976	9885	1963	Gilles
23	11 213	39 596 1321 69 108 6336	6751	1963	Gilles
24	19 937	931 144 059 27 184 2656	12003	1971	Tuckerman
25	21 701	10 659 407 14 160 5376	13066	1979	Noll y Nickel
26	23 209	81 153 7765 94 166 6816	13973	1979	Noll
27	44 497	36 509 3519 03 182 7456	26790	1979	Nelson y Stowinski
28	86 243	144 145 836 360 495 528	51024	1982	Stowinski
29	110 903	126 204 582 603 862 528	66530	1989	Coleajt y Welsh
30	132 049	131 451 295 774 550 016	79502	1983	Stowinski
31	216 091	278 327 439 843 860 128	130100	1985	Stowinski
32	766 859	15 161 8575 865 731 328	455663	1982	Stowinski y Gage
33	859 433	838 486 226 416 167 936	517430	1984	Stowinski y Gage
34	1287 787	84 073 2889 118 054 128	757293	1986	Stowinski y Gage
35	1386 269	33 182 254 723 375 16	841042	1986	Amstrong, Wollman, et al. (GIMPS)
36	2976 221	194 276 425 174 462 2976	1791964	1987	Spence, Wollman, et al. (GIMPS)
37	3021 377	81 168 848 024 457 856	1819050	1988	Clarkson, Wollman, Kurowski, et al. (GIMPS)
38	6972 593	955 178 035 123 572 736	4189190	1989	Agarwal, Wollman, Kurowski, et al. (GIMPS)
39	13 466 117	422 764 159 863 021 056	8107692	2001	Cameron, Wollman, Kurowski, et al. (GIMPS)
40	20 996 111	70 550 8909 206 896 128	1264058	2003	Shaffer, Wollman, Kurowski, et al. (GIMPS)
41	24 036 563	446 232 328 572 950 528	14471465	2004	Freidley, Wollman, Kurowski, et al. (GIMPS)
42	25 964 891	746 209 841 791 081 128	1563490	2005	Reese, Wollman, Kurowski, et al. (GIMPS)
43	30 402 457	487 437 765 164 704 256	18304103	2005	Cooper, Boone, Wollman, Kurowski, et al. (GIMPS)
44	32 582 657	779 846 855 577 120 256	19616714	2006	Cooper, Boone, Wollman, Kurowski, et al. (GIMPS)



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