

- Ej. 1:** Desarrollar un programa en el que se ingresen por teclado números enteros hasta que se hayan ingresado 5 números pares e informar por pantalla si alguno de ellos es también múltiplo de cuatro. Ejemplo:

```
Ingrese numero entero: 5
Ingrese numero entero: 13
Ingrese numero entero: 6
Numero Par. Total de numeros pares ingresados: 1
Ingrese numero entero: 9
Ingrese numero entero: 12
Numero Par. Tambien es multiplo de 4. Total de numeros pares ingresados: 2
Ingrese numero entero: 22
Numero Par. Total de numeros pares ingresados: 3
Ingrese numero entero: 14
Numero Par. Total de numeros pares ingresados: 4
Ingrese numero entero: 17
Ingrese numero entero: 20
Numero Par. Tambien es multiplo de 4. Total de numeros pares ingresados: 5

FIN
```

- Ej. 2:** Desarrollar un programa en el que se ingresen por teclado una cantidad indefinida de números enteros positivos hasta que se ingrese 0. A continuación el programa debe indicar por pantalla cuál fue el mayor y cuál el menor. Ejemplo:

```
Ingrese numeros enteros positivos (finalice con 0): 5
5
34
22
21
21
3
76
56
41
0
El mayor es 76 y el menor es 3.
```

 PONTIFICIA UNIVERSIDAD CATÓLICA ARGENTINA	Facultad de Ingeniería y Ciencias Agrarias	INFORMÁTICA GENERAL
		Practica N° 4
	Ciclos	

- Ej. 3:** Desarrollar una función que reciba como parámetro un valor numérico y determine si dicho número es primo o no, retornando verdadero (True) o falso (False) respectivamente. Luego utilizar la función en un programa que solicite al usuario el ingreso de una cantidad **cant** (número natural) y que muestre por pantalla dos listados: primero un listado de los números primos comprendidos entre 1 y **cant** y luego otro listado con los primeros **cant** números primos. Ambos listados se deben imprimir por pantalla a 10 columnas, alineando los números por las unidades. Ejemplo

Ingrese cantidad (numero natural): 57									
Primos entre 1 y 57:									
2	3	5	7	11	13	17	19	23	29
31	37	41	43	47	53				
Primeros 57 primos:									
2	3	5	7	11	13	17	19	23	29
31	37	41	43	47	53	59	61	67	71
73	79	83	89	97	101	103	107	109	113
127	131	137	139	149	151	157	163	167	173
179	181	191	193	197	199	211	223	227	229
233	239	241	251	257	263	269			

- Ej. 4:** Desarrollar una función booleana que reciba como parámetro un número entero positivo y retorne verdadero (True) o falso (False) según sea el número perfecto o no. Luego utilizarla en un programa que encuentre y muestre por pantalla los primeros cuatro números perfectos.

***Definición:** Un número perfecto es un entero positivo, que es igual a la suma de todos los enteros positivos (excluido él mismo) que son divisores del número. Por ejemplo, el primer número perfecto es 6, ya que los divisores de 6 son 1, 2, 3 y $1 + 2 + 3 = 6$.*

- Ej. 5:** Desarrollar una función booleana que reciba como parámetro un número entero de cuatro cifras y determine si el número cumple la condición de que la suma de las unidades y las centenas es igual a la suma de las decenas y las unidades de mil. Luego realizar un programa que encuentre e imprima un listado con todos los números de 4 cifras que cumplan la condición anteriormente citada. Por ejemplo, el número 7821 cumple esta condición ya que $1 + 8 = 2 + 7$.

- Ej. 6:** Desarrollar la función **aBinario** que recibe como parámetro un número decimal (*base 10, no mayor a 1000*) y retorna el número expresado en binario (*base 2*). Desarrollar un programa que ingrese por teclado un entero en base 10, invoque a la función **aBinario** y muestre por pantalla el resultado retornado por la función. Ejemplo:

Ingrese un numero decimal: 234

Numero en binario: 11101010

- Ej. 7:** Desarrollar un programa que permita ingresar las notas de una cantidad indefinida de alumnos. Considerar notas enteras en el rango de 1 a 10 e ignorar las notas no válidas (fuera el rango) ingresadas. La carga finaliza cuando la nota ingresada es 0. A continuación el programa deberá mostrar la cantidad de alumnos aplazados (nota menor a 4), la cantidad de alumnos aprobados (nota entre 4 y 7 inclusive) y la cantidad de alumnos que promocionan la materia (nota superior a 7). En cada caso, se mostrará el porcentaje del total de notas válidas cargadas que cada caso representa y el promedio general de todas las notas. Ejemplo:

Ingrese nota: 5
Ingrese nota: 4
Ingrese nota: 4
Ingrese nota: 11
Ingrese nota: 2
Ingrese nota: 8
Ingrese nota: 8
Ingrese nota: 2
Ingrese nota: 7
Ingrese nota: 9
Ingrese nota: 0

Cantidad de aplazos: 2 (22.22%)
Cantidad de aprobados: 4 (44.44%)
Cantidad de promocionados: 3 (33.33%)
Promedio general: 5.44

	Facultad de Ingeniería y Ciencias Agrarias	INFORMÁTICA GENERAL
		Practica N° 4
	Ciclos	

Ej. 8: La operación factorial de un número entero no negativo n (expresado como $n!$) es el producto que resulta de multiplicar n por todos los enteros inferiores a él hasta el uno ($0! = 1$ *por definición*). Ejemplo:

$$5! = 5 * 4 * 3 * 2 * 1 = 120$$

$$10! = 10 * 9 * 8 * 7 * 6 * 5 * 4 * 3 * 2 * 1 = 3628800$$

$$n! = n * (n-1) * (n-2) * \dots * 3 * 2 * 1 = \dots\dots\dots$$

Desarrollar un programa que solicite el ingreso de un número entero, verifique si se trata de un número mayor o igual a 0 y calcule su factorial. Para el cálculo del factorial se debe desarrollar una función que reciba como parámetro el número, realice la operatoria y retorne el resultado. En caso de que el usuario ingrese un número negativo, imprimir una advertencia. Ejemplos:

Ingrese un número entero: 5

El factorial de 5 es: 120

Ingrese un número entero: -10

No se puede calcular el factorial de un número negativo.

Ej. 9: Desarrollar una función booleana que reciba como parámetro un número entero positivo (de hasta nueve cifras) y retorne verdadero (**True**) si es capicúa o falso (**False**) en caso contrario. Un número capicúa es aquel que leído de izquierda a derecha es igual que leído de derecha a izquierda. Por ejemplo 82428 es capicúa.

Desarrollar un programa que solicite un número por teclado e informe si éste es capicúa o no según el resultado retornado por la función.

Ayuda: Para programar la función considere invertir el número y luego compararlo con el número original, si resultan iguales, entonces es capicúa.

Figuras con estructuras de repetición

ACLARACION: Todos los programas que soliciten ingresos de datos deben ser validados.

- Ej. 10:** Desarrollar un programa que solicite al usuario el ingreso de la base y la altura de un rectángulo (números naturales mayores o iguales a dos, debe validarse). Luego deberá invocar a una función que reciba como parámetros los valores de la base y altura ingresados y esta dibuje dicho rectángulo en la pantalla. Ejemplo:

```
Ingrese base: 7
Ingrese altura: -1
Error, debe ser mayor o igual a 2. Ingrese altura: 3

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

- Ej. 11:** Ídem ejercicio 10 pero dibujando un rectángulo hueco.

```
Ingrese base: 7
Ingrese altura: 4

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

- Ej. 12:** Desarrollar un programa que solicite al usuario que ingrese la base de un triángulo rectángulo de catetos iguales (número natural mayor o igual a tres, debe validarse). Luego desde el programa principal invocar a una función que dibuje en la pantalla usando asteriscos dicho triángulo usando el valor de la base pasada como parámetro. Ejemplo:

Ingrese base: 5

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

- Ej. 13:** Desarrollar un programa que solicite al usuario que ingrese una base (número entero impar mayor o igual a tres, validar el valor ingresado). El programa invocará a una función que reciba como parámetro la base ingresada y que dibuje en pantalla un triángulo isósceles de asteriscos. Ejemplo:

Ingrese base: 8
Número no válido. Ingrese base: 11

```

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

- Ej. 14:** Desarrollar un programa que solicite al usuario que ingrese la altura de un triángulo (número entero impar mayor o igual a cinco, validar el valor ingresado). Luego deberá invocar a una función que reciba como parámetro la altura ingresada y que dibuje en pantalla el triángulo de asteriscos como se puede ver en el ejemplo:

Ingrese altura: 7

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

- Ej. 15:** Desarrollar un programa que solicite al usuario que ingrese la diagonal de un rombo (validar que el valor ingresado sea impar y mayor o igual a cinco). Luego deberá invocar a una función que reciba como parámetro la diagonal ingresada y que dibuje en pantalla el rombo de asteriscos como se puede ver en el ejemplo:

Ingrese diagonal: 8

Valor incorrecto. Ingrese diagonal: 7

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

