

▷ 1. **Mínimo de 4 números**

Dados cuatro valores numéricos, se desea encontrar el menor de ellos.

- Escribe un programa que resuelva este problema.
- Escribe ahora un programa que resuelva el mismo problema pero que utilice únicamente tres comparaciones.

Solución

- ```
if n1 < n2 :
 if n1 < n3 :
 if n1 < n4 :
 min = n1
 else:
 min = n4
 else:
 if n3 < n4:
 min = n3
 else:
 min = n4
else:
 if n2 < n3:
 if n2 < n4:
 min = n2
 else:
 min = n4
 else:
 if n3 < n4:
 min = n3
 else :
 min = n4
#min es el mínimo
```
- ```
if n1<n2:
    min1=n1
else:
    min1=n2
if n3<n4:
    min2=n3
else:
    min2=n4
if min1<min2:
    min=min1
```

```

else
    min=min2
#min contiene el mínimo

```

▷ **2. Encuentra la actividad adecuada**

Escribe un programa que lea una temperatura introducida a través del teclado y muestre por pantalla la actividad más apropiada para dicha temperatura teniendo en cuenta los siguientes criterios.

| ACTIVIDAD | TEMPERATURA IDÓNEA |
|-----------|--------------------|
| Natación | temp >30 |
| Tenis | 20 <temp ≤ 30 |
| Golf | 10 <temp ≤ 20 |
| Esquí | 5 <temp ≤ 10 |
| Parchís | temp ≤ 5 |

Solución

```

def activity(temp):
    """
    This function returns the proper activity according to the given temperature.
    @type temp: int or float
    @rtype: str
    @return:
        'swimming' if temp > 30
        'tennis' if 20 < temp <= 30
        'golf' if 10 < temp <= 20
        'ski' if 5 < temp <= 10
        'ludo' if temp <= 5
    """
    if temp > 30:
        act = "swimming"
    elif 20 < temp <= 30:
        act = "tennis"
    elif 10 < temp <= 20:
        act = "golf"
    elif 5 < temp <= 10:
        act = "ski"
    else:
        act = "ludo"
    return act

```

▷ **3. Ecuación de Segundo Grado**

Escribe un programa que permita calcular las soluciones a una ecuación de segundo grado $Ax^2 + Bx + C = 0$. Obviamente, los datos del programa son los coeficientes de la ecuación A, B y C .

Solución

```

from math import sqrt
def quadratic (a, b, c):
    """

```

```

This function returns a tuple with the solution of the equation
a*x**2 + b*x + c = 0
If the real solution does not exists returns (None, None).
@param a: a != 0
@type a: float
@type b: float
@type c: float
@return: 2-element tuple of float or (None, None)
"""
disc = b**2 - 4*a*c;
if disc>=0:
    sol1 = (-b + sqrt(disc)) / (2 * a)
    sol2 = (-b - sqrt(disc)) / (2 * a)
else:
    sol1 = None
    sol2 = None
return (sol1, sol2)

```

▷ 4. Posición de un punto en un cuadrante

Escribe un programa que lea las coordenadas de un punto e indique en el cuadrante que se encuentra dicho punto.

Solución

```

def quadrant(x, y):
    """
    This function returns the cuadrant of the point x,y

    2 | 1
    |
    |
    -----
    3 | 4
    |
    |

    @type x: float
    @type y: float
    @return: 1, 2, 3 or 4 depending on the postion of (x,y)
    @rtype: int
    """
    if x>=0:
        if y>=0:
            q = 1
        else:
            q = 4
    else:
        if y>=0:
            q = 2
        else:
            q = 3
    return q

```

▷ 5. Rotación del alfabeto

Imaginemos las letras del alfabeto ordenadas y dispuestas en círculo. Esto es, a la derecha de

la *A* se encuentra la *B*, luego la *C* y así sucesivamente hasta la *Z*; a la derecha de la *Z* se encuentra nuevamente la *A*.

Definimos una rotación de longitud *n* como aquella que lleva a una determinada letra *n* posiciones hacia su derecha.

Ejemplo La rotación de longitud 1 lleva la *A* a la *B*, la *V* a la *W* y la *Z* a la *A*.

La rotación de longitud 3 lleva la *A* a la *D*, la *V* a la *Y* y la *Z* a la *C*.

Rotación de longitud 1 Escribe un programa en Python que permita calcular la rotación de longitud 1. Es decir, dada una letra del alfabeto el programa debería indicar la letra correspondiente a su rotación.

Rotación de longitud arbitraria Escribe un programa en Python que permita calcular una rotación de longitud *n* arbitraria. El valor *n* debe ser leído por el programa.

Solución

Los caracteres se tratan como números según la tabla ASCII, de esta ordenación alfabética con buenas propiedades quedan excluidas letras que sí utilizamos en castellano, como la letra Ñ. Por tanto, como decíamos al principio, sólo consideraremos letras del alfabeto inglés. No importa demasiado el número concreto que le asigna a cada carácter; si tienes curiosidad prueba el siguiente programa

```
c = raw_input()
print ord(c)
```

Rotación de longitud 1 Por tanto para calcular el siguiente carácter basta la instrucción `c = char(ord(c)+1)`, siendo *c* la variable que contiene el carácter. Sólo hay que tener cuidado con la 'Z', cuyo sucesor es el carácter '['. El programa que necesitamos es

```
c = char(ord(c)+1)
if (c > 'Z'):
    c = 'A'
```

O bien sin utilizar condicionales, utilizando las operaciones de módulo:

```
numLetras = ord('Z') - ord('A') + 1;
c1 = chr(ord('A') + ((ord(c)+1 - ord('A')) % numLetras))
```

Esta última solución se puede extender fácilmente para una rotación cualquiera

Rotación de longitud arbitraria

```
c2 = chr(ord('A') + ((ord(c)+rotacion - ord('A')) % numLetras))
```

▷ 6. ¿Me puede decir la hora?

Escribe un programa que reciba tres valores de entrada que corresponden a horas, minutos y segundos. El programa debe indicar si se trata de una hora correcta o no.

Solución

```
def isCorrectTime(hour, minute, second):
    """
    This function checks if the passed time is correct
    @type hour: int
    @type minute: int
    @type second: int
```

```
@rtype: bool
"""
return ( 0<=hour and hour<24 ) and (
        0<=minute and minute<60 ) and (
        0<=second and second<60 )
```

▷ **7. Ser o no ser bisiesto**

Hay muchas aplicaciones que requieren realizar algún tipo de operación o procesamiento con fechas, por ejemplo: el cálculo del número de días que hay entre dos fechas o la escritura del calendario de un año. En estos casos, suele ser necesario determinar si un año es o no bisiesto y ésto es, precisamente, lo que se te propone en este ejercicio. Escribe una expresión que permita determinar si un valor de tipo entero positivo puede corresponder a un año bisiesto. Se considera bisiesto un año cuyo número es divisible por cuatro excepto los años que son múltiplos de cien a no ser que lo sean de 400.

Solución

Un valor entero positivo contenido en la variable **a** puede corresponder a un año bisiesto si es cierta la siguiente expresión:

$(a \% 4 == 0) \text{ and } (a \% 100 != 0) \text{ or } (a \% 400 == 0)$