TP 1 - Computación

# Ejercicio 1

El parámetro dia\_semana es True si es un día de la semana, y el parámetro vacaciones es True si estamos de vacaciones. Dormimos si no es un día de semana o estamos de vacaciones. Regresa True si dormimos.

def dormimos(dia\_semana, vacaciones):

dormimos(False,False)→True dormimos(True,False)→False dormimos(False, True) → True

# Ejercicio 2

Tenemos dos monos, a y b, y los parámetros a\_sonriendo y b\_sonriendo indican si cada uno está sonriendo. Estamos en problemas si ambos están sonriendo o si ninguno de ellos está sonriendo. Regresa True si estamos en problemas.

def problemas\_monos(a\_sonriendo, b\_sonriendo):

problemas\_monos(True, True) → True problemas\_monos(False, False) → True problemas\_monos(True, False) → False

# Ejercicio 3

Dados dos valores int, devuelva su suma. A menos que los dos valores sean los mismos, devuelva el doble de su suma. def suma\_doble(a, b):

sum\_double(1, 2) → 3 sum\_double(3, 2) → 5 sum\_double(2, 2) → 8

# Ejercicio 4

Dado un int n, devolver la diferencia absoluta entre n y 21, excepto devolver el doble de la diferencia absoluta si n es más de 21.

diferencia21(19) → 2 diferencia21(10) → 11 diferencia21(21) → 0

# Ejercicio 5

Tenemos un loro parlante. El parámetro "hora" es la hora actual en el rango 0..23. Estamos en problemas si el loro está hablando y la hora es antes de las 7 o después de 20. Regresa True si estamos en problemas.

def problema\_loro(hablando, hora):

problema\_loro(True, 6) → True problema\_loro(True, 7) → False problema\_loro(False, 6) → False

# Ejercicio 6

Dados 2 int, a y b, devuelve True si uno es 10 o si la suma es 10.

def hacer10(a, b):

makes10(9, 10) → True makes10(9, 9) → False makes10(1, 9) → True

# Ejercicio 7

Dado un int n, devuelve True si está dentro del rango 10, 100 o 200. Nota: abs(numero) calcula el valor absoluto de un número.

def cerca\_cien(n):

near\_hundred(93) → True near\_hundred(90) → True near\_hundred(89) → False

# Ejercicio 8

Dados los valores de 2 int, devuelve True si uno es negativo y otro positivo. Excepto si el parámetro "negativo" es True , a continuación, devuelve True solo si ambos son negativos.

def pos\_negativa(a, b, negativa):

pos\_negativa(1, -1, False) → True pos\_negativa(-1, 1, False) → True pos\_negativa(-4, -5, True) → True

# Ejercicio 9

Dada una cadena, devuelva una nueva cadena donde "no" se ha agregado adelante. Sin embargo, si la cadena ya comienza con "no", devuelva la cadena sin cambios.

de no\_cadena(str):

no\_cadena('caramelo') → 'no caramelo' no\_cadena('x') → 'no x' no\_cadena('no mal') → 'no mal'

# Ejercicio 10

Dado una cadena no vacia y un int n, devolver una nueva cadena donde se ha quitado el char en el índice n. El valor de n será un índice válido de un carácter en la cadena original (es decir, n estará en el rango 0..len(str)-1 inclusive).

def carácter\_perdido(str, n):

carácter\_perdido('kitten', 1) → 'ktten' carácter\_perdido('kitten', 0) → 'itten' carácter\_perdido('kitten', 4) → 'kittn'