

Apellidos y Nombre:.....  
DNI: ..... Grupo: .....

**Informática — Examen parcial de febrero 2017**  
**Grados en Matemáticas. Grupos A, B, C, D y E**  
**Facultad de Ciencias Matemáticas, UCM**  
Tipo A

**Instrucciones:**

- El examen durará **3 horas**.
- Elegir una respuesta incorrecta en una pregunta de opción múltiple (preguntas 1–10) penalizará la nota en 0.3 puntos. Las respuestas no contestadas no penalizarán la nota. En todo caso, la puntuación mínima de la parte tipo *test* será 0.
- No se permite la utilización de apuntes, libros o cualquier otro tipo de material en el examen.

Preguntas de opción múltiple (5 puntos)

1. [0.5 pt] ¿Qué valor devolvería la llamada `check("barco", 5, 'casa')`?

```
def check(a, b, c):  
    return (b % 3 < 2) and (a[6] > c[0])
```

- (a) Se produce un error en tiempo de ejecución.
- (b) True
- (c) False
- (d) 'False'

2. [0.5 pt] ¿Qué valor devolvería la llamada `grade(2)`?

```
def grade(mark):  
    msg = 'Suspenso'  
  
    if mark >= 5 and mark < 7:  
        msg = 'Aprobado'  
    elif mark < 9:  
        msg = 'Notable'  
    else:  
        msg = 'Sobresaliente'  
  
    return msg
```

- (a) 'Suspenso'
- (b) 'Aprobado'
- (c) 'Notable'
- (d) 'Sobresaliente'

3. [0.5 pt] ¿Qué valor devolvería la llamada `range(7, 5, -1)`?

- (a) [7, 6, 5]
- (b) [7, 6]
- (c) [7, 5]
- (d) [ ]

4. [0.5 pt] ¿Qué valor devolvería la llamada `divisors(6)`?

```
def divisors(n):  
    i = 1  
    count = 0  
    while i < n:  
        if n % i == 0:  
            count = count + 1  
  
    return count
```

- (a) 4
- (b) 2
- (c) [6, 3, 2, 1]
- (d) Nada, el bucle no termina.

5. [0.5 pt] ¿Qué valor devolvería la llamada `create_image()`?

```
from PIL import Image  
  
def create_image():  
    img = Image.new("L", (50, 100), 0)  
  
    for i in range(100):  
        for j in range(50):  
            if (i - 25)**2 + (j - 50)**2 <= 20**2:  
                img.putpixel((i, j), 255)  
  
    return img
```

- (a) Una imagen de 50 píxeles de ancho y 100 de alto con fondo negro y una circunferencia blanca en el centro de la imagen.
- (b) Nada. Se produce un error del tipo *image index out of range*.
- (c) Una imagen de 100 píxeles de ancho y 50 de alto con fondo negro y una línea blanca vertical.
- (d) Nada, el bucle no termina.

6. [0.5 pt] ¿Qué valor devolvería la llamada `first_later ([1,3,5,7], [2,4,4,7,8])` ?

```
def first_later(s1, s2):
    i = 0
    while i < len(s1) and i < len(s2) and s1[i] <= s2[i]:
        i += 1
    if i < len(s1) and i < len(s2):
        result = s1[i]
    else:
        result = None
    return result
```

- (a) None
- (b) 4
- (c) 5
- (d) Error

7. [0.5 pt] ¿Qué valor devolvería la llamada `stellar_evolution (15.6)`?

```
def stellar_evolution(solar_masses):
    evolution = "none"
    if solar_masses > 8:
        evolution = "supernova - neutron star / black hole"
    if solar_masses > 0.25:
        evolution = "red giant - white dwarf - black dwarf"
    if solar_masses > 0.1:
        evolution = "blue dwarf - white dwarf - black dwarf"
    return evolution
```

- (a) 'supernova - neutron star / black hole'
- (b) 'none'
- (c) 'blue dwarf - white dwarf - black dwarf'
- (d) Fallaría al comparar números enteros y flotantes.

8. [0.5 pt] ¿Qué valor devolvería la llamada `old_car('day', 'empty', 'es')`?

```
def old_car(conditions, road_state, country):  
    lights = "no headlights"  
    if conditions == "night":  
        if road_state == "empty":  
            lights = "full beam headlights"  
        else:  
            lights = "dipped headlights"  
    if conditions == "day":  
        if country == "se":  
            lights = "no headlights"  
        if country != "no":  
            lights = "dipped headlights"  
    return lights
```

- (a) 'no headlights'
- (b) 'dipped headlights'
- (c) 'full beam headlights'
- (d) None

9. [0.5 pt] ¿Qué devolvería la invocación a la función `find_multiples(3,11)`?

```
def find_multiples(x, y):  
    i = 0  
    multiples = []  
    while i in range(y):  
        if i % x == 0:  
            multiples.append(i)  
    return multiples
```

- (a) [3,6,9]
- (b) [0,3,6,9]
- (c) Nada, el bucle no terminaría.
- (d) Nada, el programa tiene errores de sintaxis.

10. [0.5 pt] Considerando el siguiente código, ¿qué mostraría por pantalla la invocación a la función `do1(7)`?

```
def do1(a):
    b = a // 2
    c = b // 2
    do2(b, a)
    do3(a, b, c)

def do2(a, b):
    d = b * 2.0
    do3(a, b, d)

def do3(a, b, c):
    if a >= b and b >= c:
        print a-(b+c)
    else:
        print c-(a+b)
```

- (a) 4.0  
-4
- (b) 4.0  
3
- (c) 4  
-4
- (d) 3  
4.0

Rellena el código que falta (1 punto)

11. [0.5 pt] Completa el código que falta en la función `mcd` para que calcule el máximo común divisor de los números  $a$  y  $b$  (que supondremos  $\geq 0$ ). El algoritmo se basa en las siguientes premisas:  $\text{mcd}(a, b) = \text{mcd}(b, a)$ ; si  $a \geq b$ ,  $\text{mcd}(a, b) = \text{mcd}(a-b, b)$ ;  $\text{mcd}(a, 0) = a$

```
def mcd(a, b):
    while b != 0:
        _____:
        _____
    return a
```

12. [0.5 pt] Completa la función `reverse_num` que calcula el *reverso* de un número entero. El resultado es otro número entero. Por ejemplo, el reverso de 1234 es 4321 y el reverso de 12000 es 21.

```
def reverse_num(number):
    reverse = 0
    while number != 0:
        reverse = _____
        number = _____
    return reverse
```

Preguntas para desarrollar (4 puntos)

13. [2 pt] Una imagen es *oscura* cuando una proporción significativa de sus píxeles son oscuros. Si la imagen es en blanco y negro, el píxel es *oscuro* cuando tenga un valor cercano a 0. En este ejercicio se pide hacer una función, que se llamará `is_dark` que indique si una imagen es oscura o no. La función tendrá un primer parámetro `image` que hará referencia a un objeto imagen. Como “ser cercano a cero” puede ser algo subjetivo, la función tendrá un segundo parámetro `level` que indicará valor de forma que todos los inferiores a él se considerarán oscuros. Por último debemos indicar cual es la proporción de valores oscuros, para ello la función tendrá un tercer parámetro `ratio` que indicará que proporción de valores oscuros debe tener la imagen para considerarla oscura. El parámetro `ratio` es un valor en el intervalo  $(0, 1)$ . Por ejemplo, si consideramos la imagen `img` con los siguientes valores:

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17
18	19	20	21	22	23
24	25	26	27	28	29
30	31	32	33	34	35

Tendremos las siguientes respuestas:

`is_dark(img, 27, .5) → True`

`is_dark(img, 10, .5) → False`

14. [2 pt] Escribe una función `communitary(lst)` para comprobar si una lista de números enteros es *comunitaria*. Diremos que una lista es comunitaria si existe un divisor  $d > 1$  común a todos los enteros de la lista. Ejemplos:

`communitary([2, 4, 6, 8]) → True`

`communitary([2, 4, 6, 8, 3]) → False`

# Answer Key for Exam A

Preguntas de opción múltiple (5 puntos)

1. [0.5 pt] ¿Qué valor devolvería la llamada `check("barco", 5, 'casa')`?

```
def check(a, b, c):  
    return (b % 3 < 2) and (a[6] > c[0])
```

- (a) Se produce un error en tiempo de ejecución.
- (b) `True`
- (c) `False`
- (d) `'False'`

2. [0.5 pt] ¿Qué valor devolvería la llamada `grade(2)`?

```
def grade(mark):  
    msg = 'Suspenso'  
  
    if mark >= 5 and mark < 7:  
        msg = 'Aprobado'  
    elif mark < 9:  
        msg = 'Notable'  
    else:  
        msg = 'Sobresaliente'  
  
    return msg
```

- (a) `'Suspenso'`
- (b) `'Aprobado'`
- (c) `'Notable'`
- (d) `'Sobresaliente'`

3. [0.5 pt] ¿Qué valor devolvería la llamada `range(7, 5, -1)`?

- (a) `[7, 6, 5]`
- (b) `[7, 6]`
- (c) `[7, 5]`
- (d) `[ ]`

4. [0.5 pt] ¿Qué valor devolvería la llamada `divisors(6)`?

```
def divisors(n):
    i = 1
    count = 0
    while i < n:
        if n % i == 0:
            count = count + 1
        i = i + 1

    return count
```

- (a) 4
- (b) 2
- (c) [6, 3, 2, 1]
- ☒ (d) Nada, el bucle no termina.

5. [0.5 pt] ¿Qué valor devolvería la llamada `create_image()`?

```
from PIL import Image

def create_image():
    img = Image.new("L", (50, 100), 0)

    for i in range(100):
        for j in range(50):
            if (i - 25)**2 + (j - 50)**2 <= 20**2:
                img.putpixel((i, j), 255)

    return img
```

- (a) Una imagen de 50 píxeles de ancho y 100 de alto con fondo negro y una circunferencia blanca en el centro de la imagen.
- ☒ (b) Nada. Se produce un error del tipo *image index out of range*.
- (c) Una imagen de 100 píxeles de ancho y 50 de alto con fondo negro y una línea blanca vertical.
- (d) Nada, el bucle no termina.

6. [0.5 pt] ¿Qué valor devolvería la llamada `first_later([1,3,5,7], [2,4,4,7,8])`?

```
def first_later(s1, s2):
    i = 0
    while i < len(s1) and i < len(s2) and s1[i] <= s2[i]:
        i += 1
    if i < len(s1) and i < len(s2):
        result = s1[i]
    else:
        result = None
    return result
```

- (a) None
- (b) 4
- ☒ (c) 5
- (d) Error



7. [0.5 pt] ¿Qué valor devolvería la llamada `stellar_evolution (15.6)`?

```
def stellar_evolution(solar_masses):  
    evolution = "none"  
    if solar_masses > 8:  
        evolution = "supernova - neutron star / black hole"  
    if solar_masses > 0.25:  
        evolution = "red giant - white dwarf - black dwarf"  
    if solar_masses > 0.1:  
        evolution = "blue dwarf - white dwarf - black dwarf"  
    return evolution
```

- (a) 'supernova - neutron star / black hole'
- (b) 'none'
- ☒ (c) 'blue dwarf - white dwarf - black dwarf'
- (d) Fallaría al comparar números enteros y flotantes.

8. [0.5 pt] ¿Qué valor devolvería la llamada `old_car('day', 'empty', 'es')`?

```
def old_car(conditions, road_state, country):  
    lights = "no headlights"  
    if conditions == "night":  
        if road_state == "empty":  
            lights = "full beam headlights"  
        else:  
            lights = "dipped headlights"  
    if conditions == "day":  
        if country == "se":  
            lights = "no headlights"  
        if country != "no":  
            lights = "dipped headlights"  
    return lights
```

- (a) 'no headlights'
- ☒ (b) 'dipped headlights'
- (c) 'full beam headlights'
- (d) None

9. [0.5 pt] ¿Qué devolvería la invocación a la función `find_multiples (3,11)`?

```
def find_multiples(x, y):  
    i = 0  
    multiples = []  
    while i in range(y):  
        if i % x == 0:  
            multiples.append(i)  
    return multiples
```

- (a) [3,6,9]
- (b) [0,3,6,9]
- ☒ (c) Nada, el bucle no terminaría.
- (d) Nada, el programa tiene errores de sintaxis.

10. [0.5 pt] Considerando el siguiente código, ¿qué mostraría por pantalla la invocación a la función `do1(7)`?

```
def do1(a):
    b = a // 2
    c = b // 2
    do2(b, a)
    do3(a, b, c)

def do2(a, b):
    d = b * 2.0
    do3(a, b, d)

def do3(a, b, c):
    if a >= b and b >= c:
        print a-(b+c)
    else:
        print c-(a+b)
```

- (a) 4.0  
-4
- ☒ (b) 4.0  
3
- (c) 4  
-4
- (d) 3  
4.0

Rellena el código que falta (1 punto)

11. [0.5 pt] Completa el código que falta en la función `mcd` para que calcule el máximo común divisor de los números  $a$  y  $b$  (que supondremos  $\geq 0$ ). El algoritmo se basa en las siguientes premisas:  $\text{mcd}(a, b) = \text{mcd}(b, a)$ ; si  $a \geq b$ ,  $\text{mcd}(a, b) = \text{mcd}(a-b, b)$ ;  $\text{mcd}(a, 0) = a$

```
def mcd(a, b):
    while b != 0:
        if a < b:
            a, b = b, a
        a = a - b
    return a
```

12. [0.5 pt] Completa la función `reverse_num` que calcula el *reverso* de un número entero. El resultado es otro número entero. Por ejemplo, el reverso de 1234 es 4321 y el reverso de 12000 es 21.

```
def reverse_num(number):
    reverse = 0
    while number != 0:
        reverse = 10* reverse + (number % 10)
        number = number / 10
    return reverse
```

Preguntas para desarrollar (4 puntos)

13. [2 pt] Una imagen es *oscura* cuando una proporción significativa de sus píxeles son oscuros. Si la imagen es en blanco y negro, el píxel es *oscuro* cuando tenga un valor cercano a 0. En este ejercicio se pide hacer una función, que se llamará `is_dark` que indique si una imagen es oscura o no. La función tendrá un primer parámetro `image` que hará referencia a un objeto imagen. Como “ser cercano a cero” puede ser algo subjetivo, la función tendrá un segundo parámetro `level` que indicará valor de forma que todos los inferiores a él se considerarán oscuros. Por último debemos indicar cual es la proporción de valores oscuros, para ello la función tendrá un tercer parámetro `ratio` que indicará que proporción de valores oscuros debe tener la imagen para considerarla oscura. El parámetro `ratio` es un valor en el intervalo  $(0, 1)$ . Por ejemplo, si consideramos la imagen `img` con los siguientes valores:

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17
18	19	20	21	22	23
24	25	26	27	28	29
30	31	32	33	34	35

Tendremos las siguientes respuestas:

`is_dark(img, 27, .5) → True`

`is_dark(img, 10, .5) → False`

**Answer:**

```
def is_dark(img, level, ratio):
    width, height = img.size
    numDark = 0
    for x in xrange(width):
        for y in xrange(height):
            if img.getpixel((x,y)) < level:
                numDark += 1
    numPx = width * height
    print numDark, numPx
    return float(numDark)/numPx >= ratio
```

14. [2 pt] Escribe una función `communitary(lst)` para comprobar si una lista de números enteros es *comunitaria*. Diremos que una lista es comunitaria si existe un divisor  $d > 1$  común a todos los enteros de la lista. Ejemplos:

`communitary([2, 4, 6, 8]) → True`

`communitary([2, 4, 6, 8, 3]) → False`

**Answer:**

```
def communitary(lst):
    result = True
    if len(lst) > 1:
        d = mcd(lst[0], lst[1])
        i = 2
        while i < len(lst) and d != 1:
            d = mcd(d, lst[i])
            i = i + 1
        result = (d != 1)
    return result
```

Apellidos y Nombre:.....  
DNI: ..... Grupo: .....

**Informática — Examen parcial de febrero 2017**  
**Grados en Matemáticas. Grupos A, B, C, D y E**  
**Facultad de Ciencias Matemáticas, UCM**  
Tipo B

**Instrucciones:**

- El examen durará **3 horas**.
- Elegir una respuesta incorrecta en una pregunta de opción múltiple (preguntas 1–10) penalizará la nota en 0.3 puntos. Las respuestas no contestadas no penalizarán la nota. En todo caso, la puntuación mínima de la parte tipo *test* será 0.
- No se permite la utilización de apuntes, libros o cualquier otro tipo de material en el examen.

Preguntas de opción múltiple (5 puntos)

1. [0.5 pt] ¿Qué valor devolvería la llamada `check("barco", 5, 'casa')`?

```
def check(a, b, c):  
    return (b % 3 < 2) and (a[6] > c[0])
```

- (a) Se produce un error en tiempo de ejecución.
- (b) True
- (c) False
- (d) 'False'

2. [0.5 pt] ¿Qué valor devolvería la llamada `grade(2)`?

```
def grade(mark):  
    msg = 'Suspenso'  
  
    if mark >= 5 and mark < 7:  
        msg = 'Aprobado'  
    elif mark < 9:  
        msg = 'Notable'  
    else:  
        msg = 'Sobresaliente'  
  
    return msg
```

- (a) 'Suspenso'
- (b) 'Aprobado'
- (c) 'Notable'
- (d) 'Sobresaliente'

3. [0.5 pt] ¿Qué valor devolvería la llamada `range(7, 5, -1)`?

- (a) [7, 6, 5]
- (b) [7, 6]
- (c) [7, 5]
- (d) [ ]

4. [0.5 pt] ¿Qué valor devolvería la llamada `divisors(6)`?

```
def divisors(n):  
    i = 1  
    count = 0  
    while i < n:  
        if n % i == 0:  
            count = count + 1  
  
    return count
```

- (a) 4
- (b) 2
- (c) [6, 3, 2, 1]
- (d) Nada, el bucle no termina.

5. [0.5 pt] ¿Qué valor devolvería la llamada `create_image()`?

```
from PIL import Image  
  
def create_image():  
    img = Image.new("L", (50, 100), 0)  
  
    for i in range(100):  
        for j in range(50):  
            if (i - 25)**2 + (j - 50)**2 <= 20**2:  
                img.putpixel((i, j), 255)  
  
    return img
```

- (a) Una imagen de 50 píxeles de ancho y 100 de alto con fondo negro y una circunferencia blanca en el centro de la imagen.
- (b) Nada. Se produce un error del tipo *image index out of range*.
- (c) Una imagen de 100 píxeles de ancho y 50 de alto con fondo negro y una línea blanca vertical.
- (d) Nada, el bucle no termina.

6. [0.5 pt] ¿Qué valor devolvería la llamada `first_later ([1,3,5,7], [2,4,4,7,8])` ?

```
def first_later(s1, s2):
    i = 0
    while i < len(s1) and i < len(s2) and s1[i] <= s2[i]:
        i += 1
    if i < len(s1) and i < len(s2):
        result = s1[i]
    else:
        result = None
    return result
```

- (a) None
- (b) 4
- (c) 5
- (d) Error

7. [0.5 pt] ¿Qué valor devolvería la llamada `stellar_evolution (15.6)`?

```
def stellar_evolution(solar_masses):
    evolution = "none"
    if solar_masses > 8:
        evolution = "supernova – neutron star / black hole"
    if solar_masses > 0.25:
        evolution = "red giant – white dwarf – black dwarf"
    if solar_masses > 0.1:
        evolution = "blue dwarf – white dwarf – black dwarf"
    return evolution
```

- (a) 'supernova – neutron star / black hole'
- (b) 'none'
- (c) 'blue dwarf – white dwarf – black dwarf'
- (d) Fallaría al comparar números enteros y flotantes.

8. [0.5 pt] ¿Qué valor devolvería la llamada `old_car('day', 'empty', 'es')`?

```
def old_car(conditions, road_state, country):  
    lights = "no headlights"  
    if conditions == "night":  
        if road_state == "empty":  
            lights = "full beam headlights"  
        else:  
            lights = "dipped headlights"  
    if conditions == "day":  
        if country == "se":  
            lights = "no headlights"  
        if country != "no":  
            lights = "dipped headlights"  
    return lights
```

- (a) 'no headlights'
- (b) 'dipped headlights'
- (c) 'full beam headlights'
- (d) None

9. [0.5 pt] ¿Qué devolvería la invocación a la función `find_multiples(3,11)`?

```
def find_multiples(x, y):  
    i = 0  
    multiples = []  
    while i in range(y):  
        if i % x == 0:  
            multiples.append(i)  
    return multiples
```

- (a) [3,6,9]
- (b) [0,3,6,9]
- (c) Nada, el bucle no terminaría.
- (d) Nada, el programa tiene errores de sintaxis.

10. [0.5 pt] Considerando el siguiente código, ¿qué mostraría por pantalla la invocación a la función `do1(7)`?

```
def do1(a):
    b = a // 2
    c = b // 2
    do2(b, a)
    do3(a, b, c)

def do2(a, b):
    d = b * 2.0
    do3(a, b, d)

def do3(a, b, c):
    if a >= b and b >= c:
        print a-(b+c)
    else:
        print c-(a+b)
```

- (a) 4.0  
-4
- (b) 4.0  
3
- (c) 4  
-4
- (d) 3  
4.0

Rellena el código que falta (1 punto)

11. [0.5 pt] Completa el código que falta en la función `mcd` para que calcule el máximo común divisor de los números  $a$  y  $b$  (que supondremos  $\geq 0$ ). El algoritmo se basa en las siguientes premisas:  $\text{mcd}(a, b) = \text{mcd}(b, a)$ ; si  $a \geq b$ ,  $\text{mcd}(a, b) = \text{mcd}(a-b, b)$ ;  $\text{mcd}(a, 0) = a$

```
def mcd(a, b):
    while b != 0:
        _____:
        _____
    return a
```

12. [0.5 pt] Completa la función `reverse_num` que calcula el *reverso* de un número entero. El resultado es otro número entero. Por ejemplo, el reverso de 1234 es 4321 y el reverso de 12000 es 21.

```
def reverse_num(number):
    reverse = 0
    while number != 0:
        reverse = _____
        number = _____
    return reverse
```

Preguntas para desarrollar (4 puntos)



13. [2 pt] Una imagen es *oscura* cuando una proporción significativa de sus píxeles son oscuros. Si la imagen es en blanco y negro, el píxel es *oscuro* cuando tenga un valor cercano a 0. En este ejercicio se pide hacer una función, que se llamará `is_dark` que indique si una imagen es oscura o no. La función tendrá un primer parámetro `image` que hará referencia a un objeto imagen. Como “ser cercano a cero” puede ser algo subjetivo, la función tendrá un segundo parámetro `level` que indicará valor de forma que todos los inferiores a él se considerarán oscuros. Por último debemos indicar cual es la proporción de valores oscuros, para ello la función tendrá un tercer parámetro `ratio` que indicará que proporción de valores oscuros debe tener la imagen para considerarla oscura. El parámetro `ratio` es un valor en el intervalo  $(0, 1)$ . Por ejemplo, si consideramos la imagen `img` con los siguientes valores:

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17
18	19	20	21	22	23
24	25	26	27	28	29
30	31	32	33	34	35

Tendremos las siguientes respuestas:

`is_dark(img, 27, .5) → True`

`is_dark(img, 10, .5) → False`

14. [2 pt] Escribe una función `communitary(lst)` para comprobar si una lista de números enteros es *comunitaria*. Diremos que una lista es comunitaria si existe un divisor  $d > 1$  común a todos los enteros de la lista. Ejemplos:

`communitary([2, 4, 6, 8]) → True`

`communitary([2, 4, 6, 8, 3]) → False`

# Answer Key for Exam B

Preguntas de opción múltiple (5 puntos)

1. [0.5 pt] ¿Qué valor devolvería la llamada `check("barco", 5, 'casa')`?

```
def check(a, b, c):  
    return (b % 3 < 2) and (a[6] > c[0])
```

- (a) Se produce un error en tiempo de ejecución.
  - (b) `True`
  - (c) `False`
  - (d) `'False'`
2. [0.5 pt] ¿Qué valor devolvería la llamada `grade(2)`?

```
def grade(mark):  
    msg = 'Suspenso'  
  
    if mark >= 5 and mark < 7:  
        msg = 'Aprobado'  
    elif mark < 9:  
        msg = 'Notable'  
    else:  
        msg = 'Sobresaliente'  
  
    return msg
```

- (a) `'Suspenso'`
  - (b) `'Aprobado'`
  - (c) `'Notable'`
  - (d) `'Sobresaliente'`
3. [0.5 pt] ¿Qué valor devolvería la llamada `range(7, 5, -1)`?
- (a) `[7, 6, 5]`
  - (b) `[7, 6]`
  - (c) `[7, 5]`
  - (d) `[ ]`

4. [0.5 pt] ¿Qué valor devolvería la llamada `divisors(6)`?

```
def divisors(n):
    i = 1
    count = 0
    while i < n:
        if n % i == 0:
            count = count + 1
        i = i + 1

    return count
```

- (a) 4
- (b) 2
- (c) [6, 3, 2, 1]
- ☒ (d) Nada, el bucle no termina.

5. [0.5 pt] ¿Qué valor devolvería la llamada `create_image()`?

```
from PIL import Image

def create_image():
    img = Image.new("L", (50, 100), 0)

    for i in range(100):
        for j in range(50):
            if (i - 25)**2 + (j - 50)**2 <= 20**2:
                img.putpixel((i, j), 255)

    return img
```

- (a) Una imagen de 50 píxeles de ancho y 100 de alto con fondo negro y una circunferencia blanca en el centro de la imagen.
- ☒ (b) Nada. Se produce un error del tipo *image index out of range*.
- (c) Una imagen de 100 píxeles de ancho y 50 de alto con fondo negro y una línea blanca vertical.
- (d) Nada, el bucle no termina.

6. [0.5 pt] ¿Qué valor devolvería la llamada `first_later([1,3,5,7], [2,4,4,7,8])`?

```
def first_later(s1, s2):
    i = 0
    while i < len(s1) and i < len(s2) and s1[i] <= s2[i]:
        i += 1
    if i < len(s1) and i < len(s2):
        result = s1[i]
    else:
        result = None
    return result
```

- (a) None
- (b) 4
- ☒ (c) 5
- (d) Error

7. [0.5 pt] ¿Qué valor devolvería la llamada `stellar_evolution (15.6)`?

```
def stellar_evolution(solar_masses):  
    evolution = "none"  
    if solar_masses > 8:  
        evolution = "supernova – neutron star / black hole"  
    if solar_masses > 0.25:  
        evolution = "red giant – white dwarf – black dwarf"  
    if solar_masses > 0.1:  
        evolution = "blue dwarf – white dwarf – black dwarf"  
    return evolution
```

- (a) 'supernova – neutron star / black hole'
- (b) 'none'
- ☒ (c) 'blue dwarf – white dwarf – black dwarf'
- (d) Fallaría al comparar números enteros y flotantes.

8. [0.5 pt] ¿Qué valor devolvería la llamada `old_car('day', 'empty', 'es')`?

```
def old_car(conditions, road_state, country):  
    lights = "no headlights"  
    if conditions == "night":  
        if road_state == "empty":  
            lights = "full beam headlights"  
        else:  
            lights = "dipped headlights"  
    if conditions == "day":  
        if country == "se":  
            lights = "no headlights"  
        if country != "no":  
            lights = "dipped headlights"  
    return lights
```

- (a) 'no headlights'
- ☒ (b) 'dipped headlights'
- (c) 'full beam headlights'
- (d) None

9. [0.5 pt] ¿Qué devolvería la invocación a la función `find_multiples (3,11)`?

```
def find_multiples(x, y):  
    i = 0  
    multiples = []  
    while i in range(y):  
        if i % x == 0:  
            multiples.append(i)  
    return multiples
```

- (a) [3,6,9]
- (b) [0,3,6,9]
- ☒ (c) Nada, el bucle no terminaría.
- (d) Nada, el programa tiene errores de sintaxis.

10. [0.5 pt] Considerando el siguiente código, ¿qué mostraría por pantalla la invocación a la función `do1(7)`?

```
def do1(a):
    b = a // 2
    c = b // 2
    do2(b, a)
    do3(a, b, c)

def do2(a, b):
    d = b * 2.0
    do3(a, b, d)

def do3(a, b, c):
    if a >= b and b >= c:
        print a-(b+c)
    else:
        print c-(a+b)
```

- (a) 4.0  
-4
- ☒ (b) 4.0  
3
- (c) 4  
-4
- (d) 3  
4.0

Rellena el código que falta (1 punto)

11. [0.5 pt] Completa el código que falta en la función `mcd` para que calcule el máximo común divisor de los números  $a$  y  $b$  (que supondremos  $\geq 0$ ). El algoritmo se basa en las siguientes premisas:  $\text{mcd}(a, b) = \text{mcd}(b, a)$ ; si  $a \geq b$ ,  $\text{mcd}(a, b) = \text{mcd}(a-b, b)$ ;  $\text{mcd}(a, 0) = a$

```
def mcd(a, b):
    while b != 0:
        if a < b:
            a, b = b, a
        a = a - b
    return a
```

12. [0.5 pt] Completa la función `reverse_num` que calcula el *reverso* de un número entero. El resultado es otro número entero. Por ejemplo, el reverso de 1234 es 4321 y el reverso de 12000 es 21.

```
def reverse_num(number):
    reverse = 0
    while number != 0:
        reverse = 10*reverse + (number%10)
        number = number / 10
    return reverse
```

Preguntas para desarrollar (4 puntos)

13. [2 pt] Una imagen es *oscura* cuando una proporción significativa de sus píxeles son oscuros. Si la imagen es en blanco y negro, el píxel es *oscuro* cuando tenga un valor cercano a 0. En este ejercicio se pide hacer una función, que se llamará `is_dark` que indique si una imagen es oscura o no. La función tendrá un primer parámetro `image` que hará referencia a un objeto imagen. Como “ser cercano a cero” puede ser algo subjetivo, la función tendrá un segundo parámetro `level` que indicará valor de forma que todos los inferiores a él se considerarán oscuros. Por último debemos indicar cual es la proporción de valores oscuros, para ello la función tendrá un tercer parámetro `ratio` que indicará que proporción de valores oscuros debe tener la imagen para considerarla oscura. El parámetro `ratio` es un valor en el intervalo  $(0, 1)$ . Por ejemplo, si consideramos la imagen `img` con los siguientes valores:

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17
18	19	20	21	22	23
24	25	26	27	28	29
30	31	32	33	34	35

Tendremos las siguientes respuestas:

`is_dark(img, 27, .5) → True`

`is_dark(img, 10, .5) → False`

**Answer:**

```
def is_dark(img, level, ratio):
    width, height = img.size
    numDark = 0
    for x in xrange(width):
        for y in xrange(height):
            if img.getpixel((x,y)) < level:
                numDark += 1
    numPx = width * height
    print numDark, numPx
    return float(numDark)/numPx >= ratio
```

14. [2 pt] Escribe una función `communitary(lst)` para comprobar si una lista de números enteros es *comunitaria*. Diremos que una lista es comunitaria si existe un divisor  $d > 1$  común a todos los enteros de la lista. Ejemplos:

`communitary([2, 4, 6, 8]) → True`

`communitary([2, 4, 6, 8, 3]) → False`

**Answer:**

```
def communitary(lst):
    result = True
    if len(lst) > 1:
        d = mcd(lst[0], lst[1])
        i = 2
        while i < len(lst) and d != 1:
            d = mcd(d, lst[i])
            i = i + 1
        result = (d != 1)
    return result
```