



IIC2115 – Programación como Herramienta para la Ingeniería (II/2017)

Examen

Tiempo: 2 horas

Nombre: _____

Conteste en el espacio asignado para cada pregunta, si la afirmación es verdadera o falsa, argumentando su respuesta en cualquiera de los dos casos.

1. La salida del siguiente código es 0 2 7

```
class Clase:
    __variable = 0
    def suma(self, incremento):
        self.__variable += incremento
        return str(self.__variable)
objeto = Clase()
print(objeto.__variable, objeto.suma(2), objeto.suma(5))
```

Respuesta:

2. Es posible simular el comportamiento de una cola, mediante el uso de 2 stacks.

Respuesta:

3. Con el fin de visualizar de mejor manera los valores extremos de una dimensión de conjunto de datos, un analista aplica una transformación logarítmica, seguida de un *boxplot*.

Respuesta:

4. Considere el siguiente código:

```
class Clase:
    def f(self, atributos, valores):
        self.a = atributos
        self.b = valores
objeto = Clase()
objeto.f(1,2)
```

Que genera el error descrito a continuación:

```
NameError                                Traceback (most recent call last)
<ipython-input-8-7a7a98224d06> in <module>()
      4         self.b = valores
      5 objeto = Clase()
----> 6 objeto.f(1,2)

<ipython-input-8-7a7a98224d06> in f(self, atributos, valores)
      2     def f(self, atributos, valores):
      3         self.a = atributos
----> 4         self.b = valores
      5 objeto = Clase()
      6 objeto.f(1,2)

NameError: name 'valors' is not defined
```

Para corregir este error, es necesario entregar listas en vez de números enteros cuando se llama a la función f.

Respuesta:

5. La siguiente consulta retorna el tercer mayor sueldo de entre los empleados:

```
SELECT DISTINCT sueldo
FROM Empleados e1
WHERE 3 = (SELECT COUNT(DISTINCT sueldo)
          FROM Empleados e2
          WHERE e1.sueldo >= e2.sueldo)
```

Respuesta:

6. El siguiente algoritmo de ordenamiento es estable:

```
def sort(arr):
    less = []
    pivotList = []
    more = []
    if len(arr) <= 1:
        return arr
    else:
        pivot = arr[0]
        for i in arr:
            if i < pivot:
                less.append(i)
            elif i > pivot:
                more.append(i)
            else:
                pivotList.append(i)
        less = sort(less)
        more = sort(more)
        return less + pivotList + more
```

Respuesta:

7. Los siguientes dos matrices (adyacencia e incidencia, respectivamente), representan los mismos grafos:

$$G_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad G_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & -1 \end{bmatrix}$$

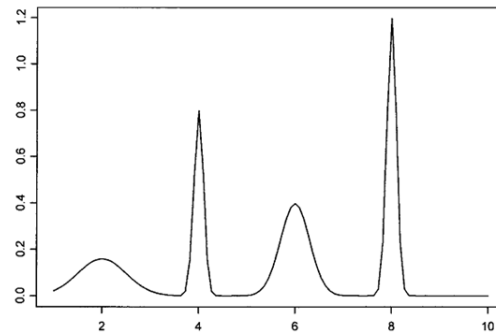
Respuesta:

8. La siguiente secuencia de comandos permite siempre actualizar el estado de un repositorio git, con todos los cambios locales realizados:

```
git add .
git commit -m "mensaje"
git push
```

Respuesta:

9. Al generar un histograma sobre los valores de una variable, se obtiene la siguiente figura:



En base a esta, es recomendable rellenar los valores faltantes utilizando la media de los valores existentes.

Respuesta:

10. Considere el siguiente código:

```
class Clase:
    numero = 0
    def f(numero):
        self.numero = numero
objeto = Clase()
objeto.f(3)
```

Que genera el error descrito a continuación:

```
TypeError                                Traceback (most recent call last)
<ipython-input-5-250ae31fd5ea> in <module>()
      4         self.numero = numero
      5 objeto = Clase()
----> 6 objeto.f(3)
```

```
TypeError: f() takes 1 positional argument but 2 were given
```

Para corregir este error, basta con quitar el argumento **3** de la llamada a la función **f**.

Respuesta:

11. Considere un árbol donde los nodos contiene un entero, con la restricción que el valor del nodo padre es mayor o que el de sus hijos. Si se busca un valor particular en este árbol, utilizando backtracking en base a la restricción descrita anteriormente, DFS siempre visitará menos nodos que BFS.

Respuesta:

12. La salida del siguiente código es ('valor', 20)

```
def tupla_valor():  
    str = ""  
    x = 20  
    return str, x;  
valor = tupla_valor()  
valor[0] = "valor"  
print(valor)
```

Respuesta:

13. Al ejecutar el siguiente código, se imprime en pantalla el string A.

```
class A:  
    def imprimir(self):  
        print("C")  
  
class B(A):  
    def imprimir(self):  
        print("A")  
  
class C(A):  
    def imprimir(self):  
        print("D")  
  
class D(C, B):  
    pass  
  
d = D()  
d.imprimir()
```

Respuesta:

14. En base al siguiente código, se puede afirmar que se realizan 14 llamadas a la función `fib` para calcular `fib(5)`:

```
def fib(n):  
    if n == 0:  
        return 0  
    elif n == 1:  
        return 1  
    else:  
        return fib(n-2) + fib(n-1)  
  
fib(5)
```

Respuesta:

15. El siguiente código en $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ permite desplegar como imagen el contenido del archivo `imagen`:

```
\begin{verbatim}  
    \begin{figure}  
        \centering  
        \includegraphics{imagen}  
    \end{figure}  
\end{verbatim}
```

Respuesta: