

**Pregunta N°20**

**IIC1103-1-6**

Se quiere tener un algoritmo que sume todos los valores de dos listas de enteros (`v1` y `v2`). Por ejemplo si `v1 = [1, 2, 3]` y `v2 = [4,5]`, total debe ser 15. Para esto, se ha escrito el programa que se muestra a continuación. Puedes asumir que `largo_v1` es el largo de la lista `v1`:

```
total = 0  
i = 0  
while i < largo_v1:  
    total = total + v1[i] + v2[i]  
    i = i + 1
```

¿Cuál de las siguientes afirmaciones es cierta para este código?

- a) El ciclo `while` termina siempre sin considerar el último elemento de cada lista, pues la condición debería ser `i <= largo_v1`, en reemplazo de `i < largo_v1` como está en el enunciado
- b) Este código no considera listas de largos distintos. Basta cambiar la condición del ciclo `while a: i <= largo_v1 or i <= largo_v2` para que funcione como se espera
- c) Este código tendrá error pues si `v1` tiene más elementos que `v2`, el índice estará fuera de rango
- d) El código anterior realizará lo que se espera en cualquier situación

**Pregunta N°21**

**IIC1103-4-2**

Observe el siguiente código donde se busca ver si un número num positivo de dos dígitos cumple una condición o no (cumple es True o False). Asuma que a//b entrega la división entera entre a y b:

```
a = num - 10 * (num//10)
```

```
b = num//10
```

```
cumple = (a+b + a*b == num)
```

¿Cuál es la condición de num para que cumple sea True?

- a) Diez veces la unidad del número más la multiplicación del número por sus dígitos debe ser igual al número
- b) La suma de las unidades más la multiplicación de ellas debe ser igual al número
- c) No hay condición, pues a siempre es 0
- d) La suma de sus dígitos más la multiplicación de ellos debe ser igual al número

**Pregunta N°22**

**Pregunta N°21**

**IIC1103**

Dado el siguiente pseudocódigo:

```
Función calculo(x) {  
    r = x;  
    i = 2*x;  
    while(i>0) {  
        r = r - x;  
        i = i - 1;  
    }  
    return r;  
}
```

¿Cuál será el resultado de calculo(a) para una variable a  $\geq 0$  de tipo int?

**Pregunta N°46**

**IIC1103-1-5 (21-2)**

A continuación, se muestra un pseudocódigo de un programa que utiliza distintos tipos de datos. Asuma que  $a \% b$  entrega el resto de la división entera entre dos enteros  $a$  y  $b$ . Por ejemplo,  $5 \% 2$  entrega el valor 1.

1.  $a = 3$
2.  $b = 15.0$
3.  $c = \text{VERDADERO}$
4.  $d = a - b$
5.  $e = d > 0$
6.  $f = b \% 2 == 1$
7.  $g = e \text{ AND } f$

Al terminar de ejecutarse el programa anterior, ¿qué valores quedan almacenados en las variables  $e$ ,  $f$  y  $g$ ?

- a)  $e = -12.0$ ,  $f = \text{FALSO}$ ,  $g = \text{FALSO}$
- b)  $e = \text{FALSO}$ ,  $f = \text{FALSO}$ ,  $g = \text{FALSO}$
- c)  $e = \text{FALSO}$ ,  $f = \text{VERDADERO}$ ,  $g = \text{FALSO}$
- d)  $e = 15.0$ ,  $f = \text{FALSO}$ ,  $g = \text{VERDADERO}$

**Pregunta N°47**  
**IIC1103-2-1-20 (21-2)**

Se tiene el siguiente pseudocódigo para generar una matriz de booleanos. Para esto, se cuenta con una lista de listas (o arreglo de arreglos) llamado `matriz` y un arreglo de números llamado `arr`. Asuma que `arr[i]` accede al elemento que se encuentra en la posición  $i$ -ésima (con  $i \geq 0$ ) de la lista o arreglo `arr`. Además, el método `arr.agregar(e)` agrega el elemento `e` al final de la lista o arreglo `arr` y que se puede crear una nueva lista o arreglo con dos elementos `a` y `b` con la sentencia `{a,b}`. Por último, la función `arr.largo()` entrega la cantidad de elementos que tiene la lista o arreglo

```
1.     i = 1
2.     WHILE (i < arr.largo()):
3.         fila = {arr[i],arr[i-1]}
4.         matriz.agregar(fila)
5.         i += 2
6.     FIN WHILE
```

Asuma que las variables `arr` y `matriz` están inicializadas. ¿Cuál de las siguientes afirmaciones es **VERDADERA**?

- a) El algoritmo puede, en algunos casos, generar una matriz irregular (no todas las filas con misma cantidad de columnas).
- b) El algoritmo genera una matriz cuadrada agregando, de manera alterna, elementos de la lista o arreglo `arr`.
- c) El algoritmo genera una matriz agregando elementos de la lista o arreglo `arr`, salvo el primero y el último. Nunca considera todos los elementos.
- d) El algoritmo genera una matriz con 2 columnas tomando todos los elementos de la lista o arreglo `arr` solo si `arr` tiene un número par de elementos.

**Pregunta N°48  
IIC1103-3-4 (22-1)**

Se quiere desarrollar un programa que identifique el sufijo más largo común en tres strings `s_1`, `s_2` y `s_3` y lo almacene en la variable `var`. Por ejemplo, entre *contar*, *recortar* y *cantar*, el sufijo más largo es *tar* pues los tres strings terminan con esos tres caracteres. Para lograr lo anterior se ha escrito el siguiente código. Recuerda que `len(s)` entrega la cantidad de caracteres del string `s` y que `s[i]` accede al carácter de `s` que se encuentra en el índice `i`. Asume que las variables `s_1`, `s_2` y `s_3` son tres variables que están inicializadas correctamente.

```
1.     var = ''  
2.     i = len(s_1) - 1  
3.     continuar = True  
4.  
5.     while i >= 0 and continuar:  
6.         if s_1[i] == s_2[i] == s_3[i]:  
7.             var = s_1[i] + var  
8.         else:  
9.             continuar = False  
10.  
11.     i -= 1
```

¿Cuál de las siguientes afirmaciones sobre este programa es **VERDADERA**?

- a) Este programa realiza lo requerido, para cualquier valor de `s_1`, `s_2` y `s_3`.
- b) Este programa encuentra el sufijo para cualquier valor de `s_1`, `s_2` y `s_3`, pero lo entrega revertido (*rat* en vez de *tar* en el ejemplo).
- c) Este programa realiza lo requerido, siempre que `s_1`, `s_2` y `s_3` tengan la misma cantidad de caracteres.
- d) Si no existe un sufijo, entonces `var` queda con el valor de `s_1`.

**Pregunta N°49**

**IIC1103-3-5 (21-2)**

A continuación, se muestra un pseudocódigo de un programa que implementa el ordenamiento por selección o *selection sort*. Asuma que `l.largo()` entrega la cantidad de elementos de la lista `l`, `l[i]` ( $i \geq 0$ ) entrega el elemento en la posición  $i$  de la lista `l`, que `encontrar_maximo(l)` entrega la posición del máximo elemento de una lista `l` y la función `intercambiar(i, j, lista)` pone el elemento de `lista[i]` en la posición  $j$  y pone en la posición  $i$  el valor que estaba originalmente en `lista[j]`. Por último, `lista[i:]` entrega la sublista que contiene los elementos desde la posición  $i$  (inclusive) en adelante.

```
1.  DEF ordenamiento_seleccion(lista)
2.      i = 0
3.
4.      WHILE i < lista.largo()
5.          max = encontrar_maximo(lista[i:])
6.          intercambiar(XX, XXX, XXXX)
7.          i += 1
8.
9.      FIN WHILE
10. fin DEF
```

Se ha perdido parte del código de la línea 6 de los parámetros que se entregan a la función `intercambiar`, ¿cuál de las siguientes alternativas muestra correctamente la llamada a la función `intercambiar` para que el código ordene la lista correctamente?

- a) `intercambiar(max + i, i, lista)`
- b) `intercambiar(lista[i], lista[max], lista)`
- c) `intercambiar(max, i, lista)`
- d) `intercambiar(max, lista[i], lista)`

**Pregunta N°46**

**IIC1103-1-1 (24-1)**

Se quiere obtener *algún* extracto de 5 dígitos de un número. El número es suficientemente grande y no contiene ceros. Este número se encuentra almacenado en la variable num.

¿Cuál de las siguientes líneas de código **NO** permite lograr el objetivo?

- a) num%100000
- b) num//100%100000
- c) (num//100%100) + (num//10000%1000)\*1000
- d) (num//100%100) + (num//10000%1000)\*100

**Pregunta N°47**  
**IIC1103-2-4 (23-2)**

A continuación, se muestra una función que recibe, como parámetro, un número n:

```
1. def funcion(n):
2.     lista = []
3.     for i in range(n):
4.         sub = [0] * n
5.         sub[i] = 1
6.         sub[n-1-i] = 1
7.         lista.append(sub)
8.     return lista
```

Asume una variable a inicializada con un valor impar mayor que 2. ¿Qué se forma queda cuando se ejecuta el siguiente código?

```
1.     l = función(a)
2.     for e in l:
3.         print(e)
```

- a) Un cuadrado de unos, con el resto ceros.
- b) Una X de unos, con el resto ceros.
- c) Un signo + de unos, con el resto ceros.
- d) Uno y ceros ( $a^2$  valores en total) intercalados irregularmente.

**Pregunta N°48**  
**IIC1103-2-7 (23-2)**

Considera el siguiente programa:

```
1.         var_1 = 3
2.         var_2 = 5
3.         var_3 = var_2 - var_1
4.         var_4 = var_2
5.         var_4 += 1
6.         res = 30
7.
8.         if var_1 >= var_2 or var_4 == 4:
9.             var_1 = 5
10.            elif var_3 < var_2:
11.                var_1 = 4
12.            elif var_1 == 4:
13.                var_1 = 6
14.            else:
15.                var_1 = 2
16.
17.            i = 0
18.            while i < var_1:
19.                res -= var_4
20.                i += 1
```

¿Qué valor queda almacenado en la variable `res` al finalizar la ejecución del programa anterior?

- a) -6
- b) 6
- c) 10
- d) 0

**Pregunta N°49**

**IIC1103-3-3 (24-1)**

Considera el siguiente programa.

Recuerda que `len(texto)` entrega la cantidad de caracteres que tiene `texto`, que los índices en un `string` comienzan en el 0, y que `texto[i]` entrega el carácter en la posición `i`.

```
1.     pal = "OXOXOXOOOXOXOXOOXOOOOOXOXO"
2.
3.     resultado = 0
4.     i = 0
5.     while i < len(pal)-1:
6.         if pal[i] == "O" and (pal[i+1] != "X" or i+1 == len(pal)):
7.             resultado += 1
8.         i += 1
```

¿Cuál es el valor de la variable `resultado` una vez que termina de ejecutarse el programa?

- a) 3
- b) 6
- c) 8
- d) 9