

[Página Principal](#) / [Mis cursos](#) / [AED \(ExFi-2022/11/24\)](#) / [Contenidos](#) / [Cuestionario Teórico \(Regulares y Promocionados\)](#)

Comenzado el jueves, 24 de noviembre de 2022, 08:12

Estado Finalizado

Finalizado en jueves, 24 de noviembre de 2022, 08:43

Tiempo empleado 31 minutos 10 segundos

Comentario - 7(siete)

Pregunta **1**

Correcta

Puntúa como 1

¿Qué significa decir que un algoritmo dado tiene un tiempo de ejecución $O(n)$?

Seleccione una:

- ☒ a. El tiempo de ejecución es lineal: si aumenta el número de datos, aumenta el tiempo en la misma proporción. ✓ ¡Correcto!
- ☐ b. A medida que aumenta el número de datos, aumenta el tiempo pero en forma muy suave: el conjunto de datos se divide en dos. se procesa una de las mitades, se desecha la otra y se repite el proceso hasta que no pueda volver a dividirse la mitad que haya quedado.
- ☐ c. El tiempo de ejecución es constante, sin importar la cantidad de datos.
- ☐ d. El proceso normalmente consiste en dos ciclos (uno dentro del otro) de aproximadamente n iteraciones cada uno, de forma que las operaciones críticas se aplican un número cuadrático de veces.

¡Correcto!

La respuesta correcta es:

El tiempo de ejecución es lineal: si aumenta el número de datos, aumenta el tiempo en la misma proporción.

Pregunta **2**


Correcta

Puntúa como 1

¿Hay algún error en el siguiente script de instrucciones en Python 3?

```
nombre = input('Nombre: ')
edad = int(input('Edad: '))
print('Datos recibidos - Nombre: ', Nombre, 'Edad: ', Edad)
```

Seleccione una:

- ☒ a. El error es que las variables *edad* y *nombre* se definieron en minúsculas al hacer la carga, y luego se usaron con  ¡Ok! mayúscula en la primera letra (*Edad* y *Nombre*) al hacer las visualizaciones.
- ☐ b. No hay ningún error.
- ☐ c. El error es que la función input() de Python 3 no puede usarse para cargar cadenas de caracteres en forma directa.
- ☐ d. El error es el uso de la función int() en la segunda carga: no existe tal función en Python 3.

¡Correcto!

La respuesta correcta es:

El error es que las variables *edad* y *nombre* se definieron en minúsculas al hacer la carga, y luego se usaron con mayúscula en la primera letra (*Edad* y *Nombre*) al hacer las visualizaciones.

Pregunta 3

Correcta

Puntúa como 1

Considere la función presentada en clases para calcular mediana entre el primero, el central y el último elemento de una partición del arreglo v delimitada por los elementos en las posiciones izq y der :

```
def get_pivot_m3(v, izq, der):  
    # calculo del pivot: mediana de tres...  
    central = int((izq + der) / 2)  
  
    if v[der] < v[izq]:  
        v[der], v[izq] = v[izq], v[der]  
  
    if v[central] < v[izq]:  
        v[central], v[izq] = v[izq], v[central]  
  
    if v[central] > v[der]:  
        v[central], v[der] = v[der], v[central]  
  
    return v[central]
```

El tamaño de la partición analizada es entonces $n = der - izq + 1$ elementos. ¿Cuál es el tiempo de ejecución de esta función, expresado en notación O y de acuerdo a ese valor de n ?

Seleccione una:

- ☐ a. $O(\log(n))$
- ☐ b. $O(n^2)$
- ☒ c. $O(1)$ ✓ ¡Correcto! Efectivamente, el cálculo no depende del tamaño de la partición, por lo que resulta de tiempo constante.
- ☐ d. $O(n)$

¡Correcto!

La respuesta correcta es:

 $O(1)$

Pregunta 4

Correcta

Puntúa como 1

Para problema general nombrado en la columna de la izquierda, seleccione la estrategia de planteo de algoritmos que se sabe haya resultado más útil para resolver ese problema, o bien la que sea que haya podido aplicarse para resolverlo aún sin llegar a una solución eficiente (considere a cada problema en su situación más general, y no casos particulares de cada uno):

Problema del Viajante.	Fuerza Bruta [$O(n!)$] / Programación Dinámica [$O(n^2 * 2^n)$]	✓
Ordenamiento rápido (Quicksort).	Divide y vencerás	✓
Problema del árbol de expansión mínimo de un grafo.	Algoritmo ávido	✓
Problema de la alineación de secuencias.	Programación dinámica	✓
Problema de las Ocho Reinas	Backtracking	✓
Generación de gráficos fractales.	Recursión	✓

¡Correcto!

La respuesta correcta es: Problema del Viajante. → Fuerza Bruta [$O(n!)$] / Programación Dinámica [$O(n^2 * 2^n)$], Ordenamiento rápido (Quicksort). → Divide y vencerás, Problema del árbol de expansión mínimo de un grafo. → Algoritmo ávido, Problema de la alineación de secuencias. → Programación dinámica, Problema de las Ocho Reinas → Backtracking, Generación de gráficos fractales. → Recursión

Pregunta 5

Incorrecta

Puntúa como 1

Suponga que se quiere desarrollar un programa para cargar una cadena de caracteres por teclado y mostrar esa cadena en la consola de salida mediante una función. ¿Cuál de las posibles respuestas *describe mejor* lo que ocurre con el programa que mostramos aquí?

```
__author__ = 'Cátedra de AED'

def mensaje(m):
    print('Mensaje:', m)

def test():
    mens = input('Ingrese un mensaje a mostrar: ')
    r = mensaje(mens)
    print('Programa terminado...')

# script principal...
test()
```

Seleccione una:

- ☐ a. El programa no compila (el intérprete lanza un error de sintaxis antes comenzar a ejecutarlo): No se puede invocar a la función *mensaje()* y asignarla en una variable, ya que esa función es de la forma sin retorno de valor.
- ☒ b. El programa compila (el intérprete no lanza error alguno al comenzar a ejecutarlo) y ejecuta, pero no muestra correctamente el mensaje esperado: en su lugar, muestra el valor *None*. ✗ Incorrecto... ¿Lo probó? Es cierto que no lanza error y ejecuta, pero el mensaje que muestra es el correcto...
- ☐ c. El programa funciona y hace exactamente lo esperado, pero la función *mensaje()* es una función sin retorno de valor, por lo que no debería ser asignada en una variable cuando se la invoca.
- ☐ d. El programa funciona, hace exactamente lo esperado, y no presenta ningún tipo de inconveniente ni elementos extraños en su código fuente.

Revise la Ficha 09, página 201 y siguientes.

La respuesta correcta es:

El programa funciona y hace exactamente lo esperado, pero la función *mensaje()* es una función sin retorno de valor, por lo que no debería ser asignada en una variable cuando se la invoca.

Pregunta 6

Incorrecta

Puntúa como 1

Analice el siguiente script básico en Python:

```
x = -9
y = 2 + x**0.5
print(y)
```

¿Cuál es el efecto de ejecutar el script anterior?

Seleccione una:

- ☐ a. Se ejecutará sin interrumpirse ni lanzar error, y la variable **y** quedará asignada con el valor 5.0 de tipo *float*.
- ☐ b. Al ejecutarse se interrumpirá en forma abrupta, lanzando un mensaje de operación matemática incorrecta (raíz cuadrada de un número negativo).
- ☒ c. Se ejecutará sin interrumpirse ni lanzar error, pero la variable **y** quedará **Incorrecto... Probó a ejecutarlo???? Efectivamente el script no provocará error, pero el resultado no será un None...** asignada con el valor *None* debido a que no puede calcular la raíz cuadrada de un número negativo.
- ☐ d. Se ejecutará sin interrumpirse ni lanzar error, y la variable **y** quedará asignada con el valor (2+3j) de tipo *complex*.

Revise la Ficha 6, página 129 y siguientes.

La respuesta correcta es:

Se ejecutará sin interrumpirse ni lanzar error, y la variable **y** quedará asignada con el valor (2+3j) de tipo *complex*.

Pregunta 7

Correcta

Puntúa como 1

¿Qué significa decir que un algoritmo dado tiene un tiempo de ejecución $O(n^2)$?

Seleccione una:

- ☐ a. A medida que aumenta el número de datos, aumenta el tiempo pero en forma muy suave: el conjunto de datos se divide en dos. se procesa una de las mitades, se desecha la otra y se repite el proceso hasta que no pueda volver a dividirse la mitad que haya quedado.
- ☐ b. El tiempo de ejecución es constante, sin importar la cantidad de datos.
- ☐ c. El tiempo de ejecución es lineal: si aumenta el número de datos, aumenta el tiempo en la misma proporción.
- ☒ d. El proceso normalmente consiste en dos ciclos (uno dentro del otro) de aproximadamente n iteraciones cada uno, **¡Correcto!** de forma que las operaciones críticas se aplican un número cuadrático de veces.

¡Correcto!

La respuesta correcta es:

El proceso normalmente consiste en dos ciclos (uno dentro del otro) de aproximadamente n iteraciones cada uno, de forma que las operaciones críticas se aplican un número cuadrático de veces.

Pregunta 8

Correcta

Puntúa como 1

¿Cuál es la cantidad de niveles del *árbol de invocaciones recursivas* que se genera al ejecutar el *Quicksort* para ordenar un arreglo de n elementos, en el **peor caso**? (Es decir: ¿Cuál es la *altura* de ese árbol en ese peor caso?)

Seleccione una:

- ☒ a. **Altura = n ✓ ¡Correcto!**
- ☐ b. Altura = $\log(n)$
- ☐ c. Altura = n^2
- ☐ d. Altura = $n * \log(n)$

¡Correcto!

La respuesta correcta es:

Altura = n

Pregunta 9

Correcta

Puntúa como 1

¿Qué hace la siguiente función en Python?

```
def check(n):  
    if n % 2 == 0:  
        return True  
    else:  
        return False
```

Seleccione una:

- ☐ a. Retorna True si el número n tomado como parámetro es impar.
- ☒ b. **Retorna True si el número n tomado como parámetro es par. ✓ ¡Ok!**
- ☐ c. Retorna True si el número n tomado como parámetro es mayor a 2.
- ☐ d. Retorna True si el número n tomado como parámetro es primo.

¡Correcto!

La respuesta correcta es:

Retorna True si el número n tomado como parámetro es par.

Pregunta **10**

Correcta

Puntúa como 1

¿Qué valor tendrá el *file pointer* del archivo representado por *m* luego de terminar de realizar las operaciones que se indican? (ignore cualquier situación de error que podría producirse al abrir el archivo o al hacer las grabaciones: sólo analice las líneas dadas suponiendo un contexto correcto):

```
import pickle
import os.path

a = 25
b = 2.876
c = 'Hola mundo'

m = open('prueba.dat', 'wb')
pickle.dump(a, m) # suponer que aquí se grabaron 5 bytes...
pickle.dump(b, m) # suponer que aquí se grabaron 12 bytes...
pickle.dump(c, m) # suponer que aquí se grabaron 20 bytes..
m.close()
```

Seleccione una:

- ☐ a. 20
- ☒ b. 37 ✓ ¡Correcto!
- ☐ c. 32
- ☐ d. 5

¡Correcto!

La respuesta correcta es: 37

Pregunta 11

Correcta

Puntúa como 1

¿Cuál de las siguientes expone correctamente la estrategia general que debe llevar a cabo un proceso de listado con filtro de los registros de un archivo, suponiendo que los registros contienen un campo de marcado lógico para gestionar eventuales bajas lógicas?

Seleccione una:

- ☐ a. 1) Abrir el archivo *m* en modo 'rb'. 2) Usar un ciclo para leer uno por uno los registros del archivo. 3) En cada vuelta del ciclo, si el registro leído está marcado como válido (*activo = True*), mostrar su contenido; en caso contrario ignorarlo (no mostrar su contenido).
- ☐ b. 1) Abrir el archivo *m* en modo 'rb'. 2) Usar un ciclo para leer uno por uno los registros del archivo. 3) En cada vuelta del ciclo, si el registro leído está marcado como válido (*activo = True*) y además cumple la condición de filtro, mostrar su contenido; en caso contrario no mostrar su contenido y detener el ciclo.
- ☐ c. 1) Abrir el archivo *m* en modo 'rb'. 2) Usar un ciclo para leer uno por uno los registros del archivo. 3) En cada vuelta del ciclo, si el registro leído está marcado como no válido (*activo = False*) y además cumple la condición de filtro, mostrar su contenido; en caso contrario ignorarlo (no mostrar su contenido).
- ☒ d. 1) Abrir el archivo *m* en modo 'rb'. 2) Usar un ciclo para leer uno por uno los registros del archivo. 3) En cada vuelta del ciclo, si el registro leído está marcado como válido (*activo = True*) y además cumple la condición de filtro, mostrar su contenido; en caso contrario ignorarlo (no mostrar su contenido). ✓ ¡Correcto!

¡Correcto!

La respuesta correcta es:

1) Abrir el archivo *m* en modo 'rb'. 2) Usar un ciclo para leer uno por uno los registros del archivo. 3) En cada vuelta del ciclo, si el registro leído está marcado como válido (*activo = True*) y además cumple la condición de filtro, mostrar su contenido; en caso contrario ignorarlo (no mostrar su contenido).

Pregunta 12

Incorrecta

Puntúa como 1

¿Cuál es la principal ventaja del borrado físico de un componente en un archivo?

Seleccione una:

- ☐ a. El archivo no ocupa más lugar que el necesario, por lo que optimiza el espacio utilizado.
- ☐ b. El borrado físico es más rápido que el marcado lógico
- ☒ c. El borrado físico garantiza que luego sean posibles los recorridos secuenciales en el archivo. ✗ Incorrecto... el archivo puede recorrerse secuencialmente, independientemente de las bajas ocasionales hayan sido realizadas en forma física o no...
- ☐ d. El borrado físico evita la necesidad de ordenar el archivo para una búsqueda rápida

Revise la Ficha de Clase 23, páginas 467 y siguientes.

La respuesta correcta es:

El archivo no ocupa más lugar que el necesario, por lo que optimiza el espacio utilizado.

Pregunta **13**

Incorrecta

Puntúa como 1

¿Qué elementos son necesarios para que una *función recursiva* se considere *bien planteada*?

Seleccione una:

- ☐ a. La función deben tener al menos una invocación a si misma en su bloque de acciones.
- ☒ b. La función debe tener al menos una invocación a si misma en su bloque de acciones, y después de esas invocaciones debe tener una o más condiciones de control que permitan interrumpir el proceso recursivo si se ha llegado a alcanzar alguna de las situaciones triviales o base del problema. ✗ Incorrecto...Si las condiciones de control están después de las invocaciones recursivas, entonces se produciría una secuencia infinita...
- ☐ c. La función debe tener al menos un ciclo en su bloque de acciones, y ese ciclo debe estar planteado de forma tal que nunca entre en un lazo infinito.
- ☐ d. La función debe tener al menos una invocación a si misma en su bloque de acciones, y antes de esas invocaciones debe tener una o más condiciones de control que permitan interrumpir el proceso recursivo si se ha llegado a alcanzar alguna de las situaciones triviales o base del problema.

Revise la Ficha 26, página 535 y siguientes.

La respuesta correcta es:

La función debe tener al menos una invocación a si misma en su bloque de acciones, y antes de esas invocaciones debe tener una o más condiciones de control que permitan interrumpir el proceso recursivo si se ha llegado a alcanzar alguna de las situaciones triviales o base del problema.

Pregunta **14**

Correcta

Puntúa como 1

¿Cuál es el valor final de la variable *res*, luego de aplicar la siguiente secuencia de instrucciones en Python? (Es recomendable que primero intente ejecutar este script y luego conteste a esta pregunta):

```
a = 20
b = 6
res = ((a // b) * 4) % 7
```

Respuesta: 

¡Correcto!

La respuesta correcta es: 5

Pregunta **15**

Incorrecta

Puntúa como 1

¿Cuáles de las siguientes propuestas generales son **ciertas** en relación al segmento de memoria conocido como **Stack Segment**?

Seleccione una o más de una:

- ☒ a. El Stack Segment funciona como una cola (o fila) de datos (modalidad **FIFO**: First In - First Out): el primer dato en llegar, se almacena al frente del stack, y será por eso el primero en ser retirado. ✗ Incorrecto... justamente se llama stack por funcionar de **otra** forma...
- ☒ b. El Stack Segment se utiliza como soporte interno en el proceso de invocación a funciones (**con o sin recursividad**): se va llenando a medida que se desarrolla la cascada de invocaciones, y se vacía a medida que se produce el proceso de regreso o vuelta atrás. ✓ ¡Correcto!
- ☐ c. El Stack Segment funciona como una pila (o apilamiento) de datos (modalidad **LIFO**: Last In - First Out): el último dato en llegar, se almacena en la cima del stack, y será por eso el primero en ser retirado.
- ☐ d. El Stack Segment se utiliza como soporte interno **solamente** en el proceso de invocación a funciones recursivas: se va llenando a medida que la cascada recursiva se va desarrollando, y se vacía a medida que se produce el proceso de regreso o vuelta atrás.

Revise la Ficha 26, página 538 y siguientes.

Las respuestas correctas son:

El Stack Segment funciona como una pila (o apilamiento) de datos (modalidad **LIFO**: Last In - First Out): el último dato en llegar, se almacena en la cima del stack, y será por eso el primero en ser retirado.,

El Stack Segment se utiliza como soporte interno en el proceso de invocación a funciones (**con o sin recursividad**): se va llenando a medida que se desarrolla la cascada de invocaciones, y se vacía a medida que se produce el proceso de regreso o vuelta atrás.

Pregunta **16**

Correcta

Puntúa como 1

El siguiente programa crea y carga un arreglo *numeros* con *n* números enteros y luego procesa ese arreglo con la función `control()`. ¿Qué hace exactamente esa función al ejecutarse?

```
__author__ = 'Cátedra de AED'

def read(numeros):
    n = len(numeros)
    for i in range(n):
        numeros[i] = int(input('Valor[' + str(i) + ']: '))


def control(numeros):
    n = len(numeros)
    for i in range(n-1):
        if numeros[i] != numeros[i+1]:
            return False
    return True

def test():
    n = int(input('Cantidad de números a cargar: '))
    numeros = n * [0]
    read(numeros)

    print()
    if control(numeros):
        print('El contenido del arreglo es correcto')
    else:
        print('El contenido del arreglo no es correcto')

if __name__ == '__main__':
    test()
```

Seleccione una:

- ☐ a. Retorna True si el arreglo está ordenado de menor a mayor.
- ☒ b. Retorna True si todos los números del arreglo son iguales.  ¡Correcto!
- ☐ c. Retorna True si el arreglo está ordenado de mayor a menor.
- ☐ d. Retorna True si los números del arreglo no son todos iguales (hay dos o más números diferentes).

¡Correcto!

La respuesta correcta es:

Retorna True si todos los números del arreglo son iguales.

Pregunta **17**

Correcta

Puntúa como 1

Suponga que se han creado y cargado por teclado dos arreglos *patentes* y *cabinas* para almacenar datos de los vehículos que se registraron en las distintas cabinas de una estación de peaje en un período dado. En el arreglo *patentes* se almacenaron cadenas que representan los números de patente de cada vehículo, y el arreglo *cabinas* se guardaron los números de las cabinas por las que cada vehículo pasó. Suponga que se pide una función *display(patentes, cabinas, x)*, que muestre sólo las patentes de los vehículos que hayan pasado por la cabina *x*. Una forma correcta de hacerlo sería la que se muestra ahora:


```
# version correcta...
def display(patentes, cabinas, x):
    exists = False
    print('Listado de vehiculos que pasaron por la cabina', x, ':')
    for i in range(len(cabinas)):
        if cabinas[i] == x:
            exists = True
            print('Patente:', patents[i])

    if not exists:
        print('No se registraron vehiculos que hayan pasado por esa cabina')
```

Suponga ahora que esa función se modifica en la forma en que se muestra más abajo, **eliminando la variable *exists* que originalmente se usaba en la función**. ¿Qué puede decirse respecto de la forma en que afecta al programa este replanteo de la función *display()*?

```
# version modificada y sugerida para este Cuestionario...
def display(patentes, cabinas, x):
    print('Listado de vehiculos que pasaron por la cabina', x, ':')
    for i in range(len(cabinas)):
        if cabinas[i] == x:
            print('Patente:', patents[i])
```

Seleccione una:

- ☐ a. El programa funcionará correctamente sin ninguna diferencia.
- ☒ b. El funcionamiento es correcto de todos modos, con una pequeña diferencia de interfaz de usuario: en la versión  ¡Correcto! original, si no había ningún vehículo que haya pasado por la cabina *x*, la función mostraba un mensaje aclarando que no había ninguno. Pero en la versión modificada, ese mensaje no aparece y el listado quedará en blanco, sólo con el título.
- ☐ c. La versión modificada no funciona correctamente: muestra siempre todos los datos, en lugar de mostrar sólo los que pasaron por la cabina *x*.
- ☐ d. La versión modificada no funciona correctamente: si no hay ningún registro para mostrar, el programa se interrumpe al finalizar el ciclo y lanza un mensaje de error en la consola.

¡Correcto!

La respuesta correcta es:

El funcionamiento es correcto de todos modos, con una pequeña diferencia de interfaz de usuario: en la versión original, si no había ningún vehículo que haya pasado por la cabina *x*, la función mostraba un mensaje aclarando que no había ninguno. Pero en la versión modificada, ese mensaje no aparece y el listado quedará en blanco, sólo con el título.

Pregunta 18

Correcta

Puntúa como 1

¿Cuál de las siguientes resume en forma correcta la idea general de la estrategia *Divide y Vencerás* para resolución de problemas?

Seleccione una:

- ☐ a. El conjunto de n datos se divide en subconjuntos de cualquier tamaño, sin importar si los tamaños de cada subconjunto coinciden entre sí. Luego se aplica recursión para procesar cada uno de esos subconjuntos. Finalmente se unen las partes que se acaban de procesar para lograr el resultado final.
- ☒ b. El conjunto de n datos se divide en subconjuntos de aproximadamente el mismo tamaño ($n/2$, $n/3$, $n/4$, etc.). Luego se aplica recursión para procesar cada uno de esos subconjuntos. Finalmente se unen las partes que se acaban de procesar para lograr el resultado final. ¡Correcto!
- ☐ c. Se aplica una regla simple que parece ser beneficiosa, sin volver atrás ni medir las consecuencias de aplicar esa regla, con la esperanza de lograr el resultado óptimo al final.
- ☐ d. Se usa una tabla para almacenar los resultados de los subproblemas que se hayan calculado, y luego cuando algún subproblema vuelve a aparecer se toma su valor desde la tabla, para evitar pérdida de tiempo.

¡Correcto!

La respuesta correcta es:

El conjunto de n datos se divide en subconjuntos de aproximadamente el mismo tamaño ($n/2$, $n/3$, $n/4$, etc.). Luego se aplica recursión para procesar cada uno de esos subconjuntos. Finalmente se unen las partes que se acaban de procesar para lograr el resultado final.

Pregunta 19

Correcta

Puntúa como 1

¿Cuál de las siguientes describe mejor la idea de funcionamiento en la que está basado el algoritmo conocido como ordenamiento por *Intercambio Directo* u *Ordenamiento de Burbuja* para ordenar de menor a mayor un arreglo de n componentes?

Seleccione una:

- ☐ a. Suponer que el arreglo tiene un subconjunto inicialmente ordenado que contiene sólo al primer elemento, luego realizar n pasadas, de forma que en cada una agregue el siguiente elemento al grupo que está ordenado.
- ☐ b. Reacomodar los n elementos del arreglo en forma aleatoria, controlar si quedó ordenado, y en caso negativo, volver a reacomodarlos en forma aleatoria, continuando así hasta que en algún momento se obtenga un arreglo ordenado...
- ☒ c. Realizar n pasadas, de forma que en cada una se compare a cada elemento con el siguiente, logrando que en cada pasada los mayores vayan acomodándose al final del arreglo. ¡Correcto!
- ☐ d. Realizar n pasadas, de forma que en cada una se determine el menor de los elementos analizados, y llevar ese menor a la casilla pivot.

¡Correcto!

La respuesta correcta es:

Realizar n pasadas, de forma que en cada una se compare a cada elemento con el siguiente, logrando que en cada pasada los mayores vayan acomodándose al final del arreglo.

Pregunta 20

Incorrecta

Puntúa como 1

Considere el programa para el *Juego del Número Secreto* que se presentó en la Ficha 8. En ese programa se usa una bandera para marcar en el algoritmo si el número secreto fue encontrado o no. Nos proponemos tratar de eliminar el uso de esa bandera y simplificar la estructura del programa, y sugerimos el que se muestra más abajo emplando una instrucción **break** para cortar el ciclo apenas se encuentre el número secreto. ¿Funciona correctamente el programa que estamos sugiriendo? *Seleccione la respuesta que mejor describa lo que está pasando con el programa propuesto.*

```
__author__ = 'Catedra de AED'

import random

print('Juego del Número Secreto... Configuración Inicial...')
limite_derecho = int(input('El número secreto estará entre 1 y: '))
cantidad_intentos = int(input('Cantidad máxima de intentos que tendrá disponible: '))

# limites iniciales del intervalo de búsqueda...
izq, der = 1, limite_derecho

# contador de intentos...
intentos = 0

# el numero secreto...
secreto = random.randint(1, limite_derecho)

# el ciclo principal... siga mientras no
# haya sido encontrado el número, y la
# cantidad de intentos no llegue a 5...
while intentos < cantidad_intentos:
    intentos += 1
    print('\nEl numero está entre', izq, 'y', der)

    # un valor para forzar al ciclo a ser [1, N]...
    num = izq - 1

    # carga y validación del número sugerido por el usuario...
    while num < izq or num > der:
        num = int(input('[Intento: ' + str(intentos) + '] => Ingrese su numero: '))
        if num < izq or num > der:
            print('Error... le dije entre', izq, 'y', der, '...')

    # controlar si num es correcto, avisar y cortar el ciclo...
    if num == secreto:
        print('\nGenio!!! Acertaste en', intentos, 'intentos')
        break

    # ... pero si no lo es, ajustar los límites
    # del intervalo de búsqueda... y seguir...
    elif num > secreto:
        der = num
    else:
        izq = num

print('\nLo siento!!! Se te acabaron los intentos. El número era:', secreto)
```

Seleccione una:

- ☒ a. Sí. Funciona correctamente para todos los casos. ✗ Incorrecto... revise los mensajes que muestran en la consola de salida cuando el jugador efectivamente adivina el número...
- ☐ b. No. No funciona bien: en la forma en que está planteado, cuando el jugador adivine el número secreto se mostrará correctamente el mensaje avisando que ganó y cortará el ciclo con la instrucción **break**. Pero como **break** corta el ciclo y no el programa completo, entonces el programa continuará e inmediatamente mostrará también el mensaje avisando que el jugador perdió, provocando ambigüedad.
- ☐ c. No. No funciona bien: en la forma en que está planteado, cuando el jugador adivine el número secreto se mostrará el mensaje avisándole que ganó pero el ciclo continuará pidiendo que se ingrese un número, hasta agotar el número de intentos disponible.

- ☐ d. No. No funciona bien: en la forma en que está planteado, se muestran en forma incorrecta los mensajes informando los límites del intervalo que contiene al número secreto en cada vuelta del ciclo, ya que las variables *izq* y *der* se actualizan en forma incorrecta.

Revise la Ficha 8, página 172 y siguientes.

La respuesta correcta es:

No. No funciona bien: en la forma en que está planteado, cuando el jugador adivine el número secreto se mostrará correctamente el mensaje avisando que ganó y cortará el ciclo con la instrucción *break*. Pero como *break* corta el ciclo y no el programa completo, entonces el programa continuará e inmediatamente mostrará también el mensaje avisando que el jugador perdió, provocando ambigüedad.

Pregunta **21**

Correcta

Puntúa como 1

En la columna de la izquierda se enumeran algunas situaciones típicas de programación. Seleccione de la columna de la derecha cuál es el tipo de ciclo que sería más adecuado o cómodo para usar en cada caso (sin que esto implique que el ciclo elegido sea obligatorio para cada caso...):

Programas controlados por menú

Ciclo do while (forzado a la forma [1, N] y conteniendo un if anidado para chequear la opción ingresada)



Recorrido de una secuencia (tupla, range, cadena, lista, etc.)

Ciclo for (iterando sobre la estructura de datos a procesar)



Esquema en el cual se sabe de antemano el número de repeticiones

Ciclo for (iterando sobre un range cuyo tamaño coincida con la cantidad de repeticiones)



Validación de valores cargados por teclado.

Ciclo while (forzado a la forma [1, N] y que incluya una condición de refuerzo de error)



Esquema en el cual se desconoce de antemano el número de repeticiones

Ciclo while (operando en forma natural como [0, n])



¡Ok!

La respuesta correcta es: Programas controlados por menú → Ciclo do while (forzado a la forma [1, N] y conteniendo un if anidado para chequear la opción ingresada), Recorrido de una secuencia (tupla, range, cadena, lista, etc.) → Ciclo for (iterando sobre la estructura de datos a procesar), Esquema en el cual se sabe de antemano el número de repeticiones → Ciclo for (iterando sobre un range cuyo tamaño coincida con la cantidad de repeticiones), Validación de valores cargados por teclado. → Ciclo while (forzado a la forma [1, N] y que incluya una condición de refuerzo de error), Esquema en el cual se desconoce de antemano el número de repeticiones → Ciclo while (operando en forma natural como [0, n])

Pregunta **22**

Correcta

Puntúa como 1

¿Cuáles de las siguientes afirmaciones son CIERTAS respecto del concepto de *reutilización de piezas de software* en programación?
[Aclaración: varias respuestas pueden ser válidas. Marque **todas** las que considere correctas]

Seleccione una o más de una:

- ☒ a. La reutilización es el proceso por el cual una misma función o módulo ya desarrollado para un programa, vuelve a usarse en otro programa en forma directa, por simple mecanismo de invocación y sin tener que modificar el nuevo programa o la propia función/módulo. ✓ ¡Ok!
- ☒ b. La reutilización permite hacer el trabajo en forma más eficiente, ordenada y profesional a un programador. ✓ ¡Ok!
- ☐ c. La reutilización es el proceso por el cual un mismo segmento de código fuente se copia y se pega en cuanto lugar el programador necesite volver a contar con dicho segmento.
- ☒ d. Los lenguajes de programación proveen grandes cantidades de funciones reutilizables en sus librerías de funciones predefinidas o estándar. ✓ ¡Ok!

¡Correcto!

Las respuestas correctas son:

La reutilización es el proceso por el cual una misma función o módulo ya desarrollado para un programa, vuelve a usarse en otro programa en forma directa, por simple mecanismo de invocación y sin tener que modificar el nuevo programa o la propia función/módulo.,

Los lenguajes de programación proveen grandes cantidades de funciones reutilizables en sus librerías de funciones predefinidas o estándar.,

La reutilización permite hacer el trabajo en forma más eficiente, ordenada y profesional a un programador.

Pregunta **23**

Correcta

Puntúa como 1

¿Por qué motivo el algoritmo Bubblesort para ordenamiento de un arreglo usa una *bandera de corte* en la versión presentada en las fichas de clase?

Seleccione una:

- ☐ a. La bandera de corte se usa para determinar si el ordenamiento debe hacerse de menor a mayor (bandera = True) o de mayor a menor (bandera = False)
- ☐ b. La bandera de corte se usa para garantizar que el arreglo quede ordenado.
- ☒ c. La bandera de corte se usa para terminar el proceso apenas se detecte que en la pasada actual no hubo intercambios, para ahorrar tiempo. ✓ ¡Correcto!
- ☐ d. No es cierto que la versión vista en clases use una bandera de corte.

¡Correcto!

La respuesta correcta es:

La bandera de corte se usa para terminar el proceso apenas se detecte que en la pasada actual no hubo intercambios, para ahorrar tiempo.

Pregunta **24**

Correcta

Puntúa como 1

¿Qué se entiende, en el contexto del Análisis de Algoritmos, por un *Orden de Complejidad*?

Seleccione una:

- ☒ a. Un conjunto o familia de funciones matemáticas que se comportan asintóticamente de la misma forma. ✓
- ☐ b. Un conjunto o familia de algoritmos que resuelven el mismo problema.
- ☐ c. Un conjunto o familia de subrutinas con similares objetivos (equivalente al concepto de *módulo*).
- ☐ d. Un conjunto de datos ordenados.

¡Correcto!



¡Correcto!

Las respuestas correctas son:

Un conjunto o familia de funciones matemáticas que se comportan asintóticamente de la misma forma.,

Un conjunto o familia de algoritmos que resuelven el mismo problema.

Pregunta **25**

Incorrecta

Puntúa como 1

¿De qué forma se representa un dato que se graba en un archivo?

Seleccione una:

- ☒ a. Se representa en sistema alfanumérico y por cada dato se utilizan tantos bytes como sea necesario para representar ese dato en forma de cadena que contenga números y caracteres. ✗
- ☐ b. Se representa en sistema decimal (base 10) y por cada dato se utilizan tantos bytes como sea necesario para representar ese dato en base 10.
- ☐ c. Se representa en formato de caracteres y por cada dato se utilizan tantos bytes como sea necesario para representar ese dato en forma de cadena de caracteres.
- ☐ d. Se representa en sistema binario (base 2) y por cada dato se utilizan tantos bytes como sea necesario para representar ese dato en base 2.

Incorrecto... La representación interna de cualquier dato en una computadora no se hace en formato de caracteres...

Revise sus notas de clase y la Ficha 22, especialmente la introducción...

La respuesta correcta es:

Se representa en sistema binario (base 2) y por cada dato se utilizan tantos bytes como sea necesario para representar ese dato en base 2.

◀ Examen Final - Registro de Calificación Final (Libres, Regulares y Promocionados)

Ir a...

Enunciado Examen Final Práctico [Regulares] ▶