

Comenzado el	domingo, 4 de junio de 2023, 17:52
Estado	Finalizado
Finalizado en	domingo, 4 de junio de 2023, 18:29
Tiempo empleado	37 minutos 18 segundos
Puntos	19/19
Calificación	10 de 10 (100%)

Pregunta **1**

Correcta

Se puntúa 1 sobre 1

En ficha referida a procesamiento de secuencias de caracteres hemos mostrado dos versiones de programas para hacerlo. En la *primera versión*, los caracteres se cargan uno por uno en una variable simple, a razón de uno por vuelta de un ciclo *while*, y en la segunda los caracteres se cargan de una sola vez en una cadena y luego se procesa esa cadena con un ciclo *for*. Si bien a nivel lógico ambas formas funcionan, ¿cuál es el inconveniente general de aplicar la *primera versión*?

Seleccione una:

- ☐ a. En realidad, no se trata de un inconveniente, sino de un error: no se puede procesar una secuencia de caracteres usando un ciclo *while*.
- ☒ b. Como los caracteres deben cargarse uno por uno con una instrucción *input()* de carga por teclado, eso provoca que por cada caracter se deba presionar <Enter>, haciendo demasiado incómodo el proceso. ✓ ¡Ok!
- ☐ c. El ciclo *while* es demasiado restrictivo e incómodo de utilizar, y siempre se prefiere un ciclo *for* en su lugar.
- ☐ d. No hay inconveniente alguno. Es tan cómoda la primera versión como la segunda.

¡Correcto!

Pregunta 2

Correcta

Se puntúa 1 sobre 1

En todos los problemas referidos a procesamiento de secuencias de caracteres que hemos analizado o propuesto, se supone que en el texto las palabras se separan entre ellas con un espacio en blanco (y solo uno), y que todo el texto finaliza con un punto. ¿Qué pasaría si en nuestros programas se chequease el final o corte de palabra con la condición que sigue (**marcada en rojo**) y sin ninguna otra condición de control de corte?

```
for car in cadena:
    # fin de palabra?
    if car == ' ':
        # final de palabra... procesar la palabra.
    else:
        # dentro de la palabra... procesar la letra.
```

Seleccione una:

- ☐ a. Provocaría que el programa se clave al llegar al final de la cadena.
- ☒ b. La última palabra del texto (que finaliza con un punto) no sería detectada ni procesada correctamente. ✓ ¡Ok!
- ☐ c. No habría problema alguno con esa condición. El programa funcionaría correctamente.
- ☐ d. Solo procesaría correctamente la última palabra del texto (que finaliza con un punto).

¡Correcto!

Pregunta 3

Correcta

Se puntúa 2 sobre 2

Suponga que se está procesando un texto carácter a carácter, y se pide detectar y contar las palabras que tengan la expresión "pi". Asuma que el flag *sp* se usa para marcar la aparición de una "p", y que el flag *spi* se usa para marcar la formación de la expresión "pi". ¿Por qué se vuelve a *False* el flag *sp* en la rama falsa de la primera condición del esquema que sigue? Asuma que este segmento de código está efectivamente dentro del ciclo *for* que se usa para recorrer la cadena que contiene al texto, y que *car* es la variable iteradora de ese *for*.

```
# deteccion de expresion "pi"...
if car == 'p':
    sp = True
else:
    if car == 'i' and sp:
        spi = True
    sp = False
```

Seleccione una:

- ☒ a. Para evitar que el flag *sp* siga en *True* si entre la "p" y la "i" llegase a entrar otra letra. ✓ ¡Ok!
- ☐ b. Para evitar que se cuente más de una vez la misma palabra si tuvo la expresión "pi".
- ☐ c. No hay razón alguna para volver a *False* ese flag. Esa instrucción hace que el esquema no funcione.
- ☐ d. Para comenzar el control de la palabra siguiente.

¡Correcto!

Pregunta 4

Correcta

Se puntúa 1 sobre 1

En todos los problemas referidos a procesamiento de secuencias de caracteres que hemos analizado o propuesto, se supone que en el texto las palabras se separan entre ellas con un espacio en blanco (y solo uno), y que todo el texto finaliza con un punto. ¿Qué pasaría si en nuestros programas se chequease el final o corte de palabra con la condición que sigue (**marcada en rojo**) y sin ninguna otra condición de control de corte?

```
for car in cadena:
    # fin de palabra?
    if car == ' ' and car == ".":
        # final de palabra... procesar la palabra.
    else:
        # dentro de la palabra... procesar la letra.
```

Seleccione una:

- ☒ a. La condición sería siempre falsa, y nunca detectaría entonces el final de ninguna palabra. ✓ ¡Ok!
- ☐ b. Provocaría que el programa se clave al llegar al final de la cadena.
- ☐ c. No habría problema alguno con esa condición. El programa funcionaría correctamente.
- ☐ d. La condición sería siempre verdadera, y nunca procesaría entonces los caracteres individuales de ninguna palabra.

¡Correcto!

Pregunta 5

Correcta

Se puntúa 2 sobre 2

Suponga que se está procesando un texto caracter a caracter, y se pide detectar y contar las palabras que tengan la expresión "mo". Asuma que el flag *sm* se usa para marcar la aparición de una "m", y que en principio se está contando en la variable *cpmo* las palabras que contienen "mo". Suponga que se propone el esquema que sigue más abajo para hacer eso.

¿Hay algún problema con ese esquema? Asuma que este segmento de código está efectivamente dentro del ciclo *for* que se usa para recorrer la cadena que contiene al texto, y que *car* es la variable iteradora de ese *for*.

```
# conteo de palabras que contienen "mo"...
if car == 'm':
    sm = True
else:
    if car == 'o' and sm:
        cpmo += 1
    sm = False
```

Seleccione una:

- ☐ a. Está mal planteado: el conteo debería ser de la forma *cpmo* += 2 en lugar de *cpmo* += 1 porque la expresión "mo" tiene dos caracteres y no solo uno.
- ☐ b. No hay ningún problema. El modelo mostrado efectivamente cuenta en la variable *cpmo* la cantidad de palabras que contienen la expresión "mo".
- ☒ c. Está mal planteado porque así como está, si la palabra contiene más de una vez la expresión "mo", el esquema contará a ✓ ¡Ok! esa palabra más de una vez.
- ☐ d. Está mal planteado porque así como está el programa **se clava**: está mal planteado el segundo *if* (no se indica contra qué valor se está comparando a la variable *sm*).

¡Correcto!

Pregunta 6

Correcta

Se puntúa 3 sobre 3

Suponga que se está procesando un texto caracter a caracter, y se pide detectar y contar las palabras que tengan una "t" en la tercera o en la quinta posición de esa palabra (por ejemplo: "antes" -una "t" en la tercera posición- o "pronto" -una "t" en la quinta- serían palabras válidas). Asuma que el contador de palabras que cumplen es la variable *ct*, y también asuma que la variable *cl* es el contador de letras de esa palabra. ¿Cuál de los siguiente modelos hace efectivamente lo pedido?

Seleccione una:

- ☐ a.

```
cl, ct, st = 0, 0, False
texto = input('Ingrese el texto (termine con un punto): ')
for car in texto:
    cl += 1
    if car == ' ' or car == '.':
        if st:
            ct += 1
        cl, st = 0, False
    else:
        if car == 't' and cl == 3 or cl == 5:
            st = True
print('Cantidad de palabras con una "t" en la tercera o en la quinta posición:', ct)
```
- ☒ b.

```
cl, ct, st = 0, 0, False
texto = input('Ingrese el texto (termine con un punto): ')
for car in texto:
    cl += 1
    if car == ' ' or car == '.':
        if st:
            ct += 1
        cl, st = 0, False
    else:
        if car == 't' and (cl == 3 or cl == 5):
            st = True
print('Cantidad de palabras con una "t" en la tercera o en la quinta posición:', ct)
```

 ✓ ¡Ok!
- ☐ c.

```
cl, ct, st = 0, 0, False
texto = input('Ingrese el texto (termine con un punto): ')
for car in texto:
    cl += 1
    if car == ' ' or car == '.':
        if st:
            ct += 1
        cl = 0
    else:
        if car == 't' and (cl == 3 or cl == 5):
            st = True
print('Cantidad de palabra con una "t" en la tercera o en la quinta posición:', ct)
```
- ☐ d.

```
cl, ct = 0, 0
texto = input('Ingrese el texto (termine con un punto): ')
for car in texto:
    cl += 1
    if car == ' ' or car == '.':
        cl = 0
    else:
        if car == 't' and (cl == 3 or cl == 5):
            ct += 1
print('Cantidad de palabras con una "t" en la tercera o en la quinta posición:', ct)
```

¡Correcto!

Pregunta 7

Correcta

Se puntúa 3 sobre 3

Suponga que se está procesando un texto caracter a caracter, y se pide detectar el porcentaje de palabras (con respecto al total de palabras del texto) que tienen al menos un dígito y al menos dos vocales. Asuma que la variable *cdv* es la que será usada para hacer ese conteo. También asuma que la variable *cp* es el contador de palabras de todo el texto y que *cl* es el contador de letras de la palabra que se está procesando. ¿Cuál de los siguiente modelos hace efectivamente lo pedido?

Seleccione una:

- ☐ a.
- ```
cp, cl, cv, cdv = 0, 0, 0, 0
sd = False
texto = input('Ingrese el texto (termine con un punto): ')
for car in texto:
 cl += 1
 if car == ' ' or car == '.':
 if cl > 1:
 cp += 1

 if sd and cv >= 2:
 cdv += 1

 cv, cl = 0, 0
 sd = False

 else:
 if car.isdigit():
 sd = True
 elif car.isalpha():
 cv += 1

porcentaje = 0.0
if cp != 0:
 porcentaje = cdv * 100 / cp
print('Porcentaje de palabras con al menos un digito y al menos dos vocales:', porcentaje)
```
- ☐ b.
- ```
cp, cl, cv, cdv = 0, 0, 0, 0
sd = False
texto = input('Ingrese el texto (termine con un punto): ')
for car in texto:
    cl += 1
    if car == ' ' or car == '.':
        if cl > 1:
            cp += 1

        if sd and cv >= 2:
            cdv += 1

        cv, cl = 0, 0
        sd = False

    else:
        if car.isdigit():
            sd = True
        elif car.upper() in 'AEIOU':
            cv += 1

porcentaje = 0.0
if cp != 0:
    porcentaje = cdv / cp
print('Porcentaje de palabras con al menos un digito y al menos dos vocales:', porcentaje)
```

☒ c.

```
cp, cl, cv, cdv = 0, 0, 0, 0
sd = False
texto = input('Ingrese el texto (termine con un punto): ')
for car in texto:
    cl += 1
    if car == ' ' or car == '.':
        if cl > 1:
            cp += 1

        if sd and cv >= 2:
            cdv += 1

        cv, cl = 0, 0
        sd = False

    else:
        if car.isdigit():
            sd = True
        elif car.upper() in 'AEIOU':
            cv += 1

porcentaje = 0.0
if cp != 0:
    porcentaje = cdv * 100 / cp
print('Porcentaje de palabras con al menos un dígito y al menos dos vocales:', porcentaje)
```

✓ ¡Ok!

☐ d.

```
cp, cl, cv, cdv = 0, 0, 0, 0
sd = False
texto = input('Ingrese el texto (termine con un punto): ')
for car in texto:
    cl += 1
    if car == ' ' or car == '.':
        if cl > 1:
            cp += 1

        if sd and cv >= 2:
            cdv += 1

        cv, cl = 0, 0
        sd = False

    else:
        if car.isdigit():
            sd = True
            if car.upper() in 'AEIOU':
                cv += 1

porcentaje = 0.0
if cp != 0:
    porcentaje = cdv * 100 / cp

print('Porcentaje de palabras con al menos un dígito y al menos dos vocales:', porcentaje)
```

¡Correcto!

Pregunta **8**

Correcta

Se puntúa 3 sobre 3

Suponga que se está procesando un texto caracter a caracter, y se pide detectar la longitud (en cantidad de caracteres) de la palabra más corta del texto. Asuma que la variable *mc* es la que contendrá ese menor valor. También asuma que la variable *cp* es el contador de palabras del texto. y que *cl* es el contador de letras de esa palabra. ¿Cuál de los siguiente modelos hace efectivamente lo pedido?

Seleccione una:

- ☐ a.
- ```
cp, cl, ct, mc = 0, 0, 0, 0
texto = input('Ingrese el texto (termine con un punto): ')
for car in texto:
 cl += 1
 if car == ' ' or car == '.':
 if cl > 1:
 cp += 1

 if cp == 1:
 mc = cl - 1
 elif cl > mc:
 mc = cl - 1

 cl = 0

print('Longitud de la palabra más corta:', mc)
```
- ☐ b.
- ```
cp, cl, ct, mc = 0, 0, 0, 0
texto = input('Ingrese el texto (termine con un punto): ')
for car in texto:
    cl += 1
    if car == ' ' or car == '.':
        if cl > 1:
            cp += 1

        if cp == 1:
            mc = cl - 1
        elif cl < mc:
            mc = cl - 1

print('Longitud de la palabra más corta:', mc)
```
- ☐ c.
- ```
cp, cl, ct, mc = 0, 0, 0, 0
texto = input('Ingrese el texto (termine con un punto): ')
for car in texto:
 cl += 1
 if car == ' ' or car == '.':
 if cl > 1:
 cp += 1

 if cp == 1:
 mc = cl
 elif cl < mc:
 mc = cl

 cl = 0

print('Longitud de la palabra más corta:', mc)
```

☒ d. 

```
cp, cl, ct, mc = 0, 0, 0, 0
texto = input('Ingrese el texto (termine con un punto): ')
for car in texto:
 cl += 1
 if car == ' ' or car == '.':
 if cl > 1:
 cp += 1

 if cp == 1:
 mc = cl - 1
 elif cl - 1 < mc:
 mc = cl - 1

 cl = 0

print('Longitud de la palabra más corta:', mc)
```

✓ ¡Ok!

¡Correcto!

Pregunta **9**

Correcta

Se puntúa 1 sobre 1

Suponga la siguiente instrucción de apertura de un archivo en Python:

```
arch = open("texto.txt", "rt")
```

¿Cuáles de las siguientes afirmaciones **son ciertas** en relación a esta instrucción? (Más de una puede ser válida, por lo que marque todas las que considere correctas)

Seleccione una o más de una:

- ☐ a. Si el archivo **texto.txt** no existe antes de ejecutar esta instrucción, la función `open()` lo crea como archivo vacío en la misma carpeta del proyecto.
- ☒ b. Si la apertura se realiza, el archivo **texto.txt** será ✓ **Correcto. La letra "t" acompañando a la "r" en el modo de apertura indica justamente eso...**
- ☐ c. Si la apertura se realiza, el archivo **texto.txt** será interpretado como un archivo binario.
- ☒ d. El archivo **texto.txt** debe existir antes de ✓ **Correcto. El archivo se está abriendo en modo "r", y por lo tanto, efectivamente, el programa se interrumpirá si ese archivo no existe antes de abrirlo.**

¡Correcto!



Pregunta **10**

Correcta


Se puntúa 2 sobre 2

Suponga la siguiente secuencia de instrucciones en Python:

```
m = open('datos.txt', 'rt')
texto = m.read()
```

Suponga que efectivamente el archivo pudo abrirse y que esa secuencia pudo ejecutarse sin problemas. ¿Cuáles de las siguientes afirmaciones **son ciertas** en relación a esta secuencia? (Más de una puede ser válida, por lo que marque todas las que considere correctas)

Seleccione una o más de una:

- ☐ a. El método `read()` leerá el primer carácter del archivo, se detendrá en ese momento, y retornará una copia de ese carácter para ser almacenada en este caso en la variable `texto`.
- ☒ b. La variable `texto` quedará asignada con una cadena que será una  **Correcto. Efectivamente, ese es el efecto de usar el método `read()` en un archivo de texto.** copia del contenido completo del archivo `datos.txt`.
- ☐ c. El método `read()` leerá el contenido del archivo hasta encontrar un espacio en blanco o un salto de línea, se detendrá en ese momento, y retornará una copia de esa cadena para ser almacenada en este caso en la variable `texto`.
- ☐ d. El método `read()` leerá los primeros diez caracteres del archivo, se detendrá en ese momento, y retornará una copia de esa cadena de diez caracteres para ser almacenada en este caso en la variable `texto`.

¡Correcto!

[◀ Materiales Adicionales para la Ficha 09](#)

Ir a...



[Simulacro 1 del Parcial 1 ▶](#)