

Paradigmas de la Programación –Primer Parcial

15 de Abril de 2025

Apellido y Nombre: _____

Ej. 1

Ej. 2

Ej. 3

Ej. 4

Ej. 5

1. En el siguiente programa, liste los valores que van tomando las siguientes variables a lo largo de la ejecución:

a) en un lenguaje con pasaje de parámetros por valor **[5 pt.]**

- i :
- A[0] :
- A[1] :
- se imprime al final :

a) en un lenguaje con pasaje de parámetros por referencia **[5 pt.]**

- i :
- A[0] :
- A[1] :
- se imprime al final :

a) en un lenguaje con pasaje de parámetros por nombre (se sustituyen los nombres de los parámetros por los nombres de los argumentos en la ejecución de la función) **[5 pt.]**

- i :
- A[0] :
- A[1] :
- se imprime al final :

```
1 int i, A[2] % declaro un escalar y un arreglo con 2 elementos
2 i <- 1
3 Procedure foo (int x, int y)
4     int temp
5     temp <- x
6     x <- y
7     i <- 0
8     y <- temp
9 end
10 A[0] <- 0
11 A[1] <- 2
12 foo(i, A[i])
13 write i, A[0], A[1]
```

2. [10 pt.] Señale con un círculo todas las opciones verdaderas. En una componente declarativa...

- a) todas las variables son locales
- b) sólo las variables libres pueden tener asignación destructiva
- c) las variables libres no pueden tener efectos secundarios
- d) la asignación de valor a una variable es necesariamente destructiva
- e) tenemos transparencia referencial, es decir, la componente junto con una asignación de valor a sus parámetros equivale semánticamente al resultado de su ejecución
- f) la asignación destructiva tiene que tener transparencia referencial

3. En el siguiente programa:

```
1 $z = 14;
2 $x = 3;
3 $y = 10;
4
5 sub foo {
6     ($x, $y) = @_; # Parametros de foo
7     $x = $y+$z;
8     $z = $x;
9 }
10
11 sub baz {
12     ($x, $y) = @_; # Parametros de baz
13     $y = $x++;
14     $z = 42;      # Variable local
15     foo($x, $y);
16     print "$z\n";
17 }
18
19 baz($x, $y);
20 print "$z\n";
```

- [5 pt.] ¿En qué línea se daría un comportamiento diferente en un lenguaje con alcance estático y en un lenguaje con alcance dinámico?

Rta.: _____

- [10 pt.] ¿Qué imprimiría el programa...

con alcance estático? Rta.: _____

con alcance dinámico? Rta.: _____

4. La siguiente expresión está mal tipada: $f(x,y) = x > y \ || \ (y \ \&\& \ x)$. Diagrame el grafo de tipado y el sistema de ecuaciones correspondiente [5 pt.], describa dónde se encuentra el problema [5 pt.] y explique cómo lo trataría un lenguaje de tipado fuerte y cómo podría tratarlo un lenguaje de tipado no fuerte [5 pt.].

5. **[10 pt.]** Indique qué se imprime en el siguiente programa...

```

1 def contar_palabras(archivo_entrada, archivo_salida):
2     try:
3         with open(archivo_entrada, 'r') as archivo:
4             text = archivo.read()
5             conteo_palabras = len(text.split())
6             with open(archivo_salida, 'w') as salida:
7                 salida.write(str(conteo_palabras))
8                 print(archivo"Conteo de palabras: {conteo_palabras}.")
9     except FileNotFoundError:
10         print("No se encontro el archivo.")
11     except FileExists:
12         print("El archivo ya existe.")
13     finally:
14         print ("Acomodamos los handlers.")
15         archivo.close()
16         salida.close()
17
18 archivo_entrada = "texto_de_entrada.txt"
19 archivo_salida = "conteo_de_palabras.txt"
20 contar_palabras(archivo_entrada, archivo_salida)
21 print ("El programa continua normalmente").

```

- [5 pt.] Si el archivo "conteo_de_palabras.txt" NO existe
- [5 pt.] Si el archivo "conteo_de_palabras.txt" existe pero el archivo "texto_de_entrada.txt" NO existe
- [5 pt.] Si tanto el archivo "conteo_de_palabras.txt" como el archivo "texto_de_entrada.txt" existen, pero ocurre un error aritmético al ejecutar `len`.