

Fundamentos de la programación y la informática

Test final

21 de junio de 2023

Grados en ingeniería aeroespacial. Turno de tarde
Universidad Rey Juan Carlos

- Ejecuta en un terminal la orden
`~mortuno/prepara`
- Comprueba que esto ha dejado en tu cuenta el fichero
`~/fpi.junio.23/test.TULOGIN.txt`, donde deberás resolver el ejercicio ¹.

Ejercicio 1 (3 puntos)

Sean a, b, c, d variables booleanas. Sean x e y variables reales. Sean las expresiones

$e1 := \text{not} (\text{not} (a \text{ or } b) \text{ and } (c \text{ and not } d));$

$e2 := \text{not} (x > 3) \text{ or } (y < 0);$

1. A partir de $e1$, escribe una expresión lógica equivalente, más legible, de nombre $s1$.
2. A partir de $e2$, escribe una expresión lógica equivalente, más legible, de nombre $s2$.
3. Escribe una expresión $n1$ que sea la negación de $s1$.
4. Escribe una expresión $n2$ que sea la negación de $s2$.

Solución

```
s1 := (a or b) or (not c or d)
s2 := (x <= 3) or (y < 0)
n1 := (not a and not b) and (c and not d)
n2 := (x > 3) and (y >= 0)
```

Ejercicio 2 (3.5 puntos)

¿Qué significa que en un lenguaje de programación *la memoria dinámica sea responsabilidad del programador*? ¿Qué alternativas hay a esto? ¿Cuáles son las ventajas e inconvenientes de cada enfoque?

Solución

Significa que el programador tiene que preocuparse de reservar memoria, liberar memoria y usar punteros. La alternativa es que sea el propio lenguaje de programación quien automáticamente reserve y libere memoria, con lo que ya no es necesario el uso de punteros.

Cuando el programador se ocupa de estas tareas, el resultado puede ser más eficiente, pero laborioso y propenso a errores. Si se hace automáticamente, todo lo contrario: menos trabajo para el programador y no habrá fallos humanos, pero los programas pueden consumir más tiempo y/o memoria.

¹TULOGIN será tu nombre de usuario en el laboratorio

Ejercicio 3 (3.5 puntos)

¿Es posible que un subprograma en Pascal devuelva más de un valor? En caso afirmativo, describe brevemente la forma o formas de conseguirlo.

Solución

Sí. Una función puede devolver un array de valores (del mismo tipo) o un registro (con valores potencialmente heterogéneos). Un procedimiento puede tener más de un parámetro de salida (anteponiendo *var* en la declaración) ²

²El compilador también permite que una función tenga parámetros de salida, aunque esto generalmente no se considera un buen diseño