

Programación Básica - Informe del Laboratorio 5

Nombre: Iker Fernández Molano Fecha: 29/10/2023



Conocimientos y capacidades adquiridas

Una vez finalizado el Laboratorio 5, considero que he realizado correctamente las siguientes tareas, y que por lo tanto estoy capacitado para repetirlas en el futuro sin que me supongan ningún problema (marcar SÍ o NO, según proceda).

		SÍ	NO
1	Saber cuándo un subprograma es una función y cuándo es un procedimiento.	X	
2	Saber cómo intercambiar información entre el programa principal y los subprogramas (paso de parámetros por copia)	X	
3	Entender el mecanismo utilizado por las funciones para devolver información al programa principal mediante la instrucción <i>return</i> .	X	
4	Entender el mecanismo utilizado por los procedimientos para devolver información al programa principal mediante parámetros de tipo OUT.	X	
5	Saber implementar pequeños subprogramas recursivos.	X	
6	Entender por qué las siguientes llamadas a subprogramas son incorrectas: 1) Considerando la función <i>cantidad_de_cifras</i> , cuya especificación es: <i>function cantidad_de_cifras (num: in integer) return integer is</i> ---Entrada: un número entero ---Pre: num:valor1 >10 ---Salida: un número entero ---Post: devolverá el número de cifras que tiene num numcifras >1 La llamada desde el programa principal es: <i>cantidad_de_cifras (1234);</i>	X	

	<p>....Indica por qué está mal.</p> <p>Al ser un function la llamada desde el programa principal debería haber sido algo parecido a:</p> <pre>Cantidad:=cantidad_de_cifras (1234);</pre> <p>Para asi meter la cantidad de cifras calculada en el subprograma en una variable, para asi poder trabajar con el valor</p> <p>2) Considerando el siguiente procedimiento:</p> <pre>procedure comprobar_el_orden(num: in integer; ascendente: out boolean; descendente: out boolean) is</pre> <p>---Entrada: un número entero</p> <p>---Pre: num:valor1 >10</p> <p>---Salida:</p> <p>---Post: si las cifras de num siguen orden ascendente, ascendente será true. Si, por el contrario, las cifras siguen orden descendente, descendente será true. Si no siguen orden ascendente ni descendente, ambas serán false.</p> <p>La llamada desde el programa principal es:</p> <pre>comprobar_el_orden(1234);</pre> <p>....Indica por qué está mal.</p> <p>Por un lado, la postcondición no es ambigua porque no especifica que cuando ascendente sea True, descendente vaya a salir, ya que es un out también como False. Por otro lado, la llamada debería haber sido algo similar a esto:</p> <pre>comprobar_el_orden(1234,es_ascendente,es_descendente);</pre> <p>Ya que en las llamadas con procedure se escriben tanto los in como los out dentro de los paréntesis.</p>	X
--	--	---

7	<p>Dada la siguiente función:</p> <pre><i>function cantidad_de_cifras(num: in integer) return integer is cont:integer:=0; begin loop exit when num=0; cont:=cont+1; num:=num/10; end loop; return(cont); end cantidad_de_cifras;</i></pre> <p>....Indica por qué el código es incorrecto.</p> <p>Por un lado, en ADA no se puede inicializar una variable antes del begin, sólo puede declararse, es decir, cont:integer:=0; daría error.</p> <p>Por otro lado, num es una constante que no puede variar, como su nombre indica, así que el num:=num/10; es incorrecto. Para que esto funcionase debería de crearse otra variable de tipo integer, inicializarla con el valor de num, y a partir de ahí, sí, ir trabajando con ella.</p>	
8	<p>OTROS: Indica cuáles</p>	

Si has respondido **NO** a alguna de las cuestiones anteriores, indica a continuación por qué, así como las medidas que vas a tomar para solventar el problema que supone no ser capaz de realizar dichas tareas trabajadas en el laboratorio.



Solución del laboratorio 5

En la entrega de la solución del laboratorio 5 que subáis a eGela, deberéis **obligatoriamente** adjuntar los ficheros fuente en un archivo comprimido (por ejemplo .zip). El nombre de este archivo comprimido deberá respetar las reglas de nombrado que se aplican a las entregas de laboratorio, esto es: InicialApellido_inicialApellido_lab5.zip.