

Programación Básica - Laboratorio 1 Introducción a la algorítmica

Nombre: Iker Fernández Molano Fecha: 11/09/2023

Ejercicio 1

Intercambiar: Pedir al usuario dos valores enteros mayores que 0 y pensar un algoritmo tal que intercambie ambos valores, de forma que el primer valor que el usuario haya introducido aparezca en pantalla en segundo lugar, y el valor que el usuario tecleó después aparezca en primer lugar.

1) Especificación

¿Es ambiguo el enunciado? ¿Por qué?

Sí. Porque el enunciado no especifica qué ocurre con las variables, no explica si éstas también se intercambian o no.

Rellenar

Entrada: Dos valores

Pre: Valor_1 y valor_2 enteros y >0

Salida. Dos valores

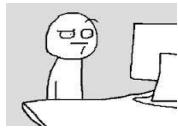
Post: En la pantalla aparecerán primero valor_2 y luego valor_1 y también se intercambiarán en memoria (num1:valor_2 y num2:valor_1)

2) Casos de Prueba¹

Num1	Num2
7	4
5	5
2	1
3	6

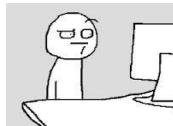
Pantalla
4 7
5 5
1 2
6 3

¹ El número de filas de esta tabla no pretende ser en absoluto indicativo del número de casos de prueba que haya.



3) Algoritmo

```
num1, num2, aux: Integer;  
escribir (“Teclea dos números enteros mayores que 0”);  
leer(num1);  
leer(num2);  
  
aux ← num1;  
num1 ← num2;           Intercambiar num1 y num2  
num2 ← aux;  
  
escribir (num1,num2);
```



Ejercicio 2

Ordenar dos números: Pedir al usuario dos valores enteros mayores que 0 y pensar un algoritmo tal que ordene esos valores de mayor a menor y los presente por pantalla (utilizar como base el algoritmo de calcular el máximo que se explicó en la clase de teoría).

1. Especificación

¿Es ambiguo el enunciado? ¿Por qué?

Sí. Porque el enunciado no especifica qué ocurre con las variables, no explica si éstas también se intercambian siguiendo el orden de los valores o no.

Rellenar

Entrada: Dos valores

Pre: Valor_1 y valor_2 enteros >0

Salida. Dos valores

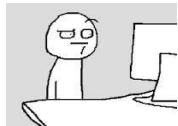
Post: En la pantalla aparecerá primero el valor mayor (valor_1 o valor_2) y luego el valor menor (valor_1 o valor_2) y también se intercambiarán en memoria, tomando el valor mayor la variable num1 y el valor menor la variable num2

1. Casos de prueba²

Num1	Num2
3	6
4	2
1	1

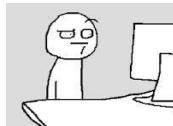
Pantalla
6 3
4 2
1 1

² El número de filas de esta tabla no pretende ser en absoluto indicativo del número de casos de prueba que haya.



2. Algoritmo

```
num1, num2, aux: Integer;  
escribir (“Teclea dos números enteros mayores que 0”);  
leer(num1);  
leer(num2);  
  
si num1 < num2 entonces;  
aux ← num1;  
num1 ← num2;           Ordenar num1 y num2  
num2 ← aux;  
fin_si;  
  
escribir (num1, num2);
```



Ejercicio 3

Ordenar tres números: Pedir al usuario tres valores enteros mayores que 0 y pensar un algoritmo tal que ordene esos valores de mayor a menor y los presente por pantalla.

1. Especificación

¿Es ambiguo el enunciado? ¿Por qué?

Sí. Porque el enunciado no especifica qué ocurre con las variables, no explica si éstas también se intercambian siguiendo el orden de los valores o no.

Rellenar

Entrada: Tres valores

Pre: Valor_1, valor_2 y valor_3 enteros y >0

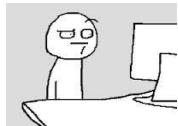
Salida. Tres valores

Post: En la pantalla aparecerá primero el valor mayor (valor_1, valor_2 o valor_3), luego el valor menor que el primero, pero mayor que el tercero (valor_1, valor_2 o valor_3) y finalmente, el valor menor de entre los tres (valor_1, valor_2 o valor_3); y también se intercambiarán en memoria, tomando, respectivamente, el valor mayor la variable num1, el valor intermedio la variable num2 y el valor menor de entre los tres la variable num3

1) Casos de prueba³

Num1	Num2	Num3	Pantalla
3	2	1	3 2 1
3	1	2	3 2 1
3	3	1	3 3 1
3	1	3	3 3 1
3	1	1	3 1 1
1	3	1	3 1 1
1	1	3	3 3 1
1	3	3	1 1 1
1	1	1	

³ El número de filas de esta tabla no pretende ser en absoluto indicativo del número de casos de prueba que haya.



2. Algoritmo

```
num1, num2, num3, aux: Integer;  
escribir (“Teclea tres números enteros mayores que 0”);  
leer(num1);  
leer(num2);  
leer(num3);  
  
si num1 < num2 entonces;  
aux ← num1;  
num1 ← num2;           Ordenar num1 y num2  
num2 ← aux;  
fin_si;  
  
si num2 < num3 entonces;  
aux ← num2;  
num2 ← num3;           Ordenar num2 y num3  
num3 ← aux;  
fin_si;  
  
si num1 < num2 entonces;  
aux ← num1;  
num1 ← num2;           Ordenar num1 y num2  
num2 ← aux;  
fin_si;  
  
escribir (num1, num2, num3);
```