

Programación Básica - Laboratorio 2

Condicionales

Nombre: Iker Fernández y María Fernández **Fecha:** 18/09/2023

1º ejercicio

ONCE: Dado el número premiado en el sorteo de la ONCE (supongamos que solo tiene 4 dígitos) y dado el número introducido por el usuario (también de 4 dígitos), hacer un programa que nos diga si el número ha sido premiado y con cuánto dinero.

4 dígitos iguales 100000 euros
3 últimos dígitos iguales 50000 euros
2 últimos dígitos iguales 3 euros

1) Especificación

¿Es ambiguo el enunciado? ¿Por qué?

Sí, porque el enunciado no especifica si el número del usuario debe estar en una misma caja o en este caso, 4 distintas.

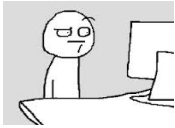
Rellenar

Entrada: 1 número entero

Pre: valor_1, valor_2, valor_3, valor_4 ≥ 0 y ≤ 9

Salida: Un texto

Post: En la pantalla aparecerá el premio que variará dependiendo de los valores (valor_1, valor_2, valor_3, valor_4) dados por el usuario y un texto.



2) Casos de prueba

Entrada1 (usuario)	Entrada2 (Premio)	Salida
4540	4540	Zorionak!! ha ganado 100000 euros
4540	5540	Zorionak!! ha ganado 50000 euros
1123	4856	Tu número no ha sido premiado, el número premiado es 4856
5556	4856	Has ganado 3 euros, el número premiado es 4856
4851	4856	Tu número no ha sido premiado, el número premiado es 4856

3) Algoritmo

N_Premio, N_Usuario: Integer;

Escribir (“Teclea tu número de lotería respectivamente en su caja”);

N_Premio ← 4856;

Leer (N_Usuario);

Si N_Usuario rem 100 = N_Premio rem 100 entonces;

 Si N_Usuario rem 1000 = N_Premio rem 1000 entonces;

 Si N_Usuario = N_Premio entonces;

 Escribir (“Has ganado 100000 euros”);

 Si no;

 Escribir (“Has ganado 50000 euros, el número premiado es 4856”);

 Fin_si;

 Si no;

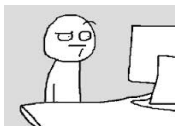
 Escribir (“Has ganado 3 euros, el número premiado es 4856”);

 Fin_si;

Si no;

 Escribir (“Tu número no ha sido premiado, el número premiado es 4856”);

Fin_si;



2º ejercicio

Segundo anterior: Escribir un programa que solicite la hora al usuario, es decir le solicitará que introduzca, la hora, los minutos y los segundos que conforman la hora. El programa deberá calcular e imprimir por pantalla esa hora pero con un segundo menos.

Es decir, si el usuario introduce

Hora 10

Min 30

Seg 40

El programa debería sacar por pantalla 10:30:39

1) Especificación

¿Es ambiguo el enunciado? ¿Por qué?

Sí, porque no indica si hay que especificar si es el mismo día o no.

Rellenar

Entrada: 3 números enteros

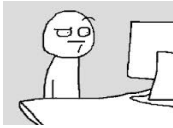
Pre: hora ($\text{valor_1y2} \geq 0$ y ≤ 23 , minuto ($\text{valor_3y4} \geq 0$ y ≤ 59 y segundo ($\text{valor_5y6} \geq 0$ y ≤ 59

Salida: 3 números enteros y True/False

Post: En la pantalla aparecerán tres valores (hora, min, seg), donde a la variable “seg” se le habrá restado 1; la variable “min” podrá variar dependiendo del valor que se le haya dado a éste y, en otros casos, también se verá contagiada la variable “hora”. También aparece una variable (día) que será “True” menos cuando las variables hora, min y seg sean todas 00, que cambiará a “False”.

2) Casos de prueba

Entrada	Salida	
Hora 10 min 30 seg 10	Segundo anterior: 10:30:09	Día: el mismo
Hora 10 min 59 seg 10	Segundo anterior: 10:59:09	Día: el mismo
Hora 00 min 00 seg 00	Segundo anterior: 23:59:59	Día: el anterior
Hora 13 min 00 seg 00	Segundo anterior: 12:59:59	Día: el mismo
Hora 13 min 36 seg 00	Segundo anterior: 13:35:59	Día: el mismo
Hora 23 min 59 seg 59	Segundo anterior: 23:59:58	Día: el mismo



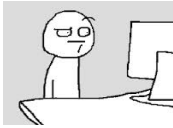
3) Algoritmo

(Conviene que numeréis las líneas de código, para luego poder hacer referencia a ellas en el apartado de simulación)

```
1. Hora, min, seg: Integer;
2. Dia: Boolean;
3. Escribir ("Escribe una hora (0-23), minuto (0-59) y segundo (0-59)");
4. Leer (hora);
5. Leer (min);
6. Leer (seg);
7. Dia ← True;
8.
9. si seg=00 entonces;
10.     si min=00 entonces;
11.         Si hora=00 entonces;
12.             hora ← hora+23;
13.             min ← min+59;
14.             seg ← seg+59;
15.             Dia ← false;
16.         Si no;
17.             hora ← hora-1;
18.             min ← min+59;
19.             seg ← seg+59;
20.         fin_si;
21.     Si no;
22.         min ← min-1;
23.         seg ← seg+59;
24.     Fin_si;
25. si no;
26.     seg ← seg-1;
27. fin_si;
28. Escribir (Hora:min:seg      Mismo día: Dia);
```

4) Simulación

Entrada	Salida	Escribir exclusivamente los números de línea del algoritmo que se ejecutan para el caso de prueba correspondiente.	Pantalla
10:30:10	10:30:09	1-9 y 25-28	10:30:09 Mismo día: True



3º ejercicio

Añadir duración a una tarea: Escribir un programa que dada la hora de comienzo de una tarea y su duración en segundos, calcule la hora en la que finalizará la tarea. La duración de la tarea puede ser de varias horas, nunca superior a media jornada (≤ 14400 segundos). Una jornada laboral será de 8 horas, comienza a las 8:00 y finaliza a las 16:00.

1) Especificación

¿Es ambiguo el enunciado? ¿Por qué?

No.

Rellenar

Entrada: 4 números enteros

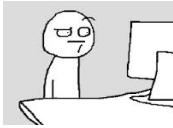
Pre: $\text{valor}_1 < 23$, valor_2 y $\text{valor}_3 < 59$ y $\text{valor}_4 > 0$ y ≤ 14400

Salida: 3 números enteros y un booleano True/False

Post: En la pantalla aparecerán tres valores (hora, min, seg), a “seg” se le habrá sumado el valor_4 y “hora” y “min” variarán de manera lógica dependiendo del valor de éste. Además, cuando $\text{hora} > 16$, “min” y “seg” se mantienen igual, pero “hora” suma a 8 la cantidad de números > 16 que había.

2) Casos de prueba

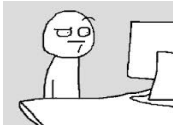
Entrada1(hora de comienzo)	Entrada2(duración)	Salida	Día siguiente
hora 10 min 30 seg 30	30 segundos	hora 10 min 31 seg 00	False
hora 10 min 30 seg 30	1800 segundos	hora 11 min 00 seg 30	False
hora 10 min 40 seg 30	3606 segundos	hora 11 min 40 seg 36	False
hora 15 min 50 seg 10	3600 segundos	hora 8 min 50 seg 10	True
hora 15 min 50 seg 10	14400 segundos	hora 11 min 50 seg 10	True
hora 11 min 40 seg 03	50 segundos	hora 11 min 40 seg 53	False
hora 16 min 00 seg 00	7200 segundos	hora 10 min 00 seg 00	True
hora 15 min 30 seg 40	3645 segundos	hora 8 min 31 seg 25	True
hora 8 min 0 seg 0	3700 segundos	hora 9 min 1 seg 40	False



3) Algoritmo

(Conviene que numeréis las líneas de código, para luego poder hacer referencia a ellas en el apartado de simulación)

1. Hora, min, seg, duracion_tarea, hora_aux, min_aux, seg_aux, min_aux2, hora_aux2: Integer;
2. Dia: Boolean;
3. Escribir (“Teclea la hora de inicio de la tarea (Hora:Minuto:Segundo) y la duración (Segundos)”)
4. Leer (hora);
5. Leer (min);
6. Leer (seg);
7. Leer (duracion_tarea);
8. Dia←True;
9. seg←seg+duracion_tarea;
10. si seg>60 entonces
11. min_aux←seg/60;
12. seg←seg rem 60;
13. min←min+min_aux;
14. si min>60 entonces
15. hora_aux← min/60;
16. min←min rem 60;
17. hora←hora+hora_aux;
18. si hora >16 entonces
19. hora_aux2←hora-16;
20. hora←8;
21. hora←hora+hora_aux2;
22. Dia←False;
23. fin_si;
24. fin_si;
25. fin_si;
26. escribir (hora:min:seg “Mismo día:” Dia);



4) Simulación

horaComienzo	duración	Salida	Escribir exclusivamente los números de línea del algoritmo que se ejecutan para el caso de prueba correspondiente	Pantalla
hora 15 min 30 seg 40	3645 segundos	hora 8 min 31 seg 25	1-26	08:31:25 Mismo día: False

4º ejercicio

Pedir al usuario que introduzca un día, un mes y un año. Escribir un algoritmo que calcule el día anterior.

- Sin tener en cuenta los años bisiestos
- Teniendo en cuenta que existen años bisiestos. Para que un año sea bisiesto deberá ser múltiplo de 4 y no ser múltiplo de 100. Los múltiplos de 400 son una excepción, ya que aun siendo múltiplos de 100 son bisiestos.

1) Especificación

¿Es ambiguo el enunciado? ¿Por qué?

Sí, porque no especifica que ocurriría antes del año 0, si habría que señalar los años A.C. o no.

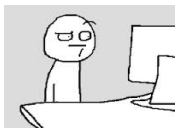
Rellenar

Entrada: 3 números enteros

Pre: Valor_1 >0 y ≤31, Valor_2 >0 y ≤12, Valor_3 > 0000

Salida: 3 números enteros

Post: En pantalla aparecerá la fecha indicada con un día menos, si la variable dia_U =1, entonces variará también el mes_U y si éste también es =1, variará año_U. Por otro lado, los años bisiestos, cuando el año sea múltiplo de 4, pero no de 100, como excepción los múltiplos de 400, tomarán el mes 2 con 29 días.

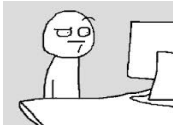


2) Casos de prueba

Entrada	Salida
día 1 mes 1 anno 2006	día 31 mes 12 anno 2005
día 1 mes 3 anno 2006	día 28 mes 2 anno 2006
día 1 mes 3 anno 2000	día 29 mes 2 anno 2000
día 1 mes 5 anno 2006	día 30 mes 4 anno 2006
día 18 mes 7 anno 2010	día 17 mes 7 anno 2010
día 29 mes 2 anno 2024	día 28 mes 2 anno 2024
día 1 mes 12 anno 2024	día 30 mes 11 anno 2024

3) Algoritmo

```
1. Día_U, mes_U, año_U, día_salida, mes_salida, año_salida: Integer;
2. Escribir ("Escribe una fecha dd/mm/aaaa");
3. Leer (día_U);
4. Leer (mes_U);
5. Leer (año_U);
6.
7. Si día_u≠1 entonces
8.     Día_salida←Día_U-1;
9.     Mes_salida←Mes_U;
10.    Año_salida←Año_U;
11. Si no
12.     Si mes_U=5 o 7 o 10 o 12 entonces
13.         Día_salida ←30;
14.         Mes_salida ←mes_U-1;
15.         Año_salida←Año_U;
16.     Fin_si;
17.     Si mes_U=2 o 4 o 6 o 8 o 9 o 11 entonces
18.         Día_salida ←31;
19.         Mes_salida ←mes_U -1;
20.         Año_salida←Año_U;
21.     Fin_si;
22.     Si mes_U=1 entonces
23.         Día_salida←31;
24.         Mes_salida←12;
25.         Año_salida←año_U-1;
26.     Fin_si;
27.     Si mes_U=3 entonces
28.         Año_salida←Año_U;
29.         Si año_U rem 4≠0 y año_U rem 100=0 y año_U rem 400≠0 entonces
30.             Día_salida←28;
31.             Mes_salida ←mes_U -1;
32.         Fin_si;
33.     Si no
```

```
34.                      Dia_salida←29;
35.                      Mes_salida ←mes_U -1;
36.                      Fin_si;
37.      Fin_si;
38. Fin_si;
39.
40. Escribir (día_salida"/"mes_salida"/"año_salida);
```

4) Simulación (como las tablas anteriores)

Entrada	Salida	Escribir exclusivamente los números de línea del algoritmo que se ejecutan para el caso de prueba correspondiente	Pantalla
dia 01 mes 03 año 2024	dia 29 mes 02 año 2024	1-7, 11, 27-28, 33-40	29/02/2024