

Guía de Ejercicios 3

Introducción

Esta guía trabajamos los elementos de control de flujo `if`, `else` y `else if` pero utilizando expresiones booleanas que usen `and` (`&&`), `or` (`||`) y `not` (`!`).

Debajo de la mayoría de los problema encontrarás una tabla con ejemplos de entrada y salidas que deben producir tus soluciones. Usa estos ejemplos para verificar que tu solución sea correcta.

Ejercicios: `if`, `else`, `else if`

1. Escribe un programa que pida dos números enteros al usuario. Si ambos números son mayores a 0, debe decir si el primer número es divisible en el segundo.

Entrada	Salida
4 2	El número 4 es divisible en el 2
5 2	El número 5 no es divisible en el 2
0 1	Existe un valor igual a 0
1 -1	Existe un valor negativo
0 -1	Existe un valor igual a 0 y otro negativo

2. Escribe un programa que reciba dos números enteros y prediga el signo que tendrá la multiplicación de ambos números.

Entrada	Salida
1 7	El resultado será positivo
1 -7	El resultado será negativo

3. Constanza quiere viajar a Japón. Para esto debe ser mayor de edad o ir en compañía de un adulto, tener un carnet de identidad y no tener antecedentes policiales. Haga un programa que

pregunte a Constanza su edad, si va acompañada de un adulto, si tiene carnet de identidad y si tiene antecedentes policiales. El programa debe decidir si Constanza puede viajar a Japón.

Ejemplo 1:

Edad de Constanza:

19

Viaja con un adulto?

no

Tiene carnet de identidad?

sí

Tiene antecedentes policiales?

no

¡Puede viajar!

Ejemplo 2:

Edad de Constanza:

12

Viaja con un adulto?

sí

Tiene carnet de identidad?

sí

Tiene antecedentes policiales?

no

¡Puede viajar!

Ejemplo 3:

Edad de Constanza:

12

Viaja con un adulto?

no

Tiene carnet de identidad?

sí

Tiene antecedentes policiales?

no

No Puede viajar :(

4. Constanza logró viajar y mientras espera el avión decide pasar a una tienda de hamburguesas. Esta tienda vende hamburguesas a \$3.000, papas fritas a \$1.000 y bebidas a \$1.000. Si un cliente compra una hamburguesa con papas o una hamburguesa con bebida tiene un 5 % de descuento, mientras que si compra hamburguesas con papas y bebida tiene un 10 % de descuento. Haga un programa que le pregunte a Constanza qué desea llevar y calcule el total de la compra.

Ejemplo 1:

Desea hamburguesa?

sí

Lleva papas?

no

Lleva bebida?

sí

El total es: \$3.800

Ejemplo 2:

Desea hamburguesa?

sí

Lleva papas?

sí

Lleva bebida?

sí

El total es: \$4.500

Ejemplo 3:

Desea hamburguesa?

no

Lleva papas?

no

Lleva bebida?

no

El total es: \$0

5. Escribe un programa que le pida al usuario 3 números. El programa debe decidir si el primer número está entre los otros dos o si es igual a uno de ellos. Es importante notar que no es importante el orden en que se ingresen los últimos dos números.

Entrada	Salida
1 -1 2	1 está entre -1 y 2
0 2 -1	1 está entre 2 y -1
10 2 3	10 no está entre 2 y 3
1 1 2	1 es igual a uno de los números

6. Escriba un programa que reciba la calificación de un ramo. Debe comprobar que la calificación sea un número entre 1.0 y 7.0. Si la calificación es menor a 4.0 debe imprimir **insuficiente**. Si la calificación es mayor o igual a 4.0 pero menor que 5.0 debe imprimir **regular**. Si la calificación es mayor o igual a 5.0 pero menor que 6.0 debe imprimir **suficiente**. Si la calificación es mayor

o igual a 6.0 pero menor que 7.0 debe imprimir **muy bien**. Finalmente, si la calificación es igual a 7.0, debe imprimir **Excelente**.

Entrada	Salida
-1.3	Calificación no válida
2.5	Insuficiente
5.5	Suficiente
7.0	Excelente

7. Escribe un programa que reciba los tres largos de los lados de un triángulo (suponga que efectivamente corresponden a lados posibles de un triángulo). Imprima si el triángulo es equilátero, isósceles o escaleno.

Entrada	Salida
1 1 1	Equilátero
3 3 2	Isósceles
3 4 5	Escaleno

8. Tres segmentos de recta pueden formar un triángulo si la suma del largo de dos de ellas es siempre mayor que el largo del tercer segmento. Por ejemplo si los largos son $a = 3$, $b = 4$ y $c = 5$ se tiene que:

- $a + b > c$
- $b + c > a$
- $c + a > b$

por lo tanto aquellos segmentos pueden formar un triángulo. Sin embargo, si tenemos que los largos son $a = 2$, $b = 1$ y $c = 1$ se tiene que:

- $a + b > c$
- $c + a > b$
- no se cumple que $b + c > a$

por lo tanto aquellos segmentos no pueden formar un triángulo. Haga un programa que reciba el largo de tres segmentos y determine si pueden formar un triángulo y si es de tipo escaleno, isósceles o equilátero.

Entrada	Salida
1	No es posible formar un triángulo
1	
2	
3	Es posible formar un triángulo isóceles
3	
2	

9. En la programación de videojuegos es muy común requerir detectar colisiones entre diferentes elementos. Detectar colisiones es simplemente revisar si dos elementos se están tocando. Sin embargo, tener un detector de colisiones perfecto es demasiado costoso en términos computacionales. Por ejemplo, en un juego 2D si queremos detectar la colisión entre elementos de 100x100 pixeles cuadrados, tendríamos que comparar cada pixel del primer elemento con cada pixel del segundo. Estas son 100 millones de comparaciones. Esto es un número alto para un computador (o una consola) considerando que debe realizar esto decena de veces por segundo y con cientos o miles de elementos.

Una técnica muy popular para detectar colisiones son los **hitbox**. La idea es representar el área de los elementos con una figura más simple para la cual sea más fácil detectar las colisiones. En los juegos 2D se suelen usar rectángulos, tal como lo muestra la figura.

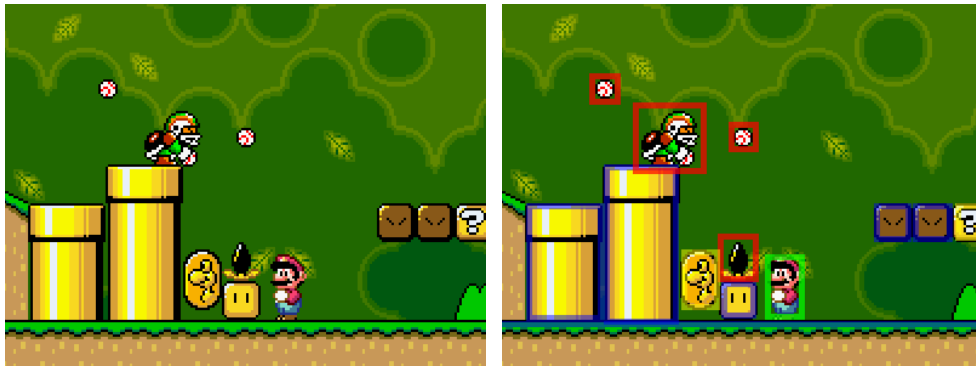


Figura 1: Ejemplo de hitbox rectangulares en Super Mario World

En este ejercicio deberas programar un detector de colisiones usando hitbox rectangulares para dos elementos. Tu programa debe recibir la posición x, posición y, ancho y alto de cada rectángulo y debe indicar si hay colisión. Considera que la posición (x,y) está en el centro del rectángulo.

Entrada	Salida
0 0 30 40 90 0 60 60	Mario[x, y, ancho, alto]: Enemigo[x, y, ancho, alto]: Mario está a salvo
55 0 30 40 90 0 60 60	Mario[x, y, ancho, alto]: Enemigo[x, y, ancho, alto]: Mario muere
65 40 30 40 90 0 60 60	Mario[x, y, ancho, alto]: Enemigo[x, y, ancho, alto]: Mario muere
135 50 30 40 90 0 60 60	Mario[x, y, ancho, alto]: Enemigo[x, y, ancho, alto]: Mario está a salvo
125 -20 30 40 90 0 60 60	Mario[x, y, ancho, alto]: Enemigo[x, y, ancho, alto]: Mario muere

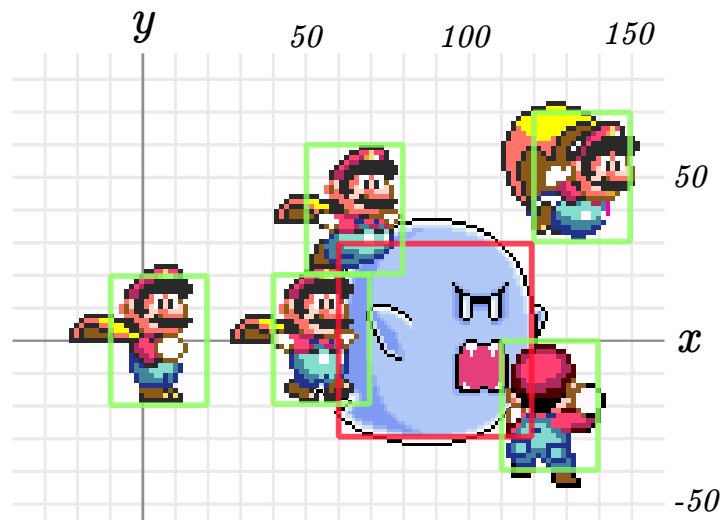


Figura 2: Diagrama que muestra los casos de prueba

10. En este ejercicio haremos un programa para generar el nombre de una pizza. El usuario pide una pizza con tres ingredientes. Los ingredientes se pueden repetir. Lo primero que debe mostrar el programa es el menú con los posibles ingredientes:

```
== Ingredientes ==
1. Queso
2. Tomate
3. Aceituna
4. Jamón
5. Pepperoni
6. Champiñones
```

Luego, el usuario debe ingresar los tres ingredientes. Hecho esto, el programa debe desplegar el nombre de la pizza. Para el nombre de la pizza existen ciertas reglas que se deben cumplir:

- Dependiendo de los ingredientes, el nombre debe comenzar con un prefijo:
 - Si tiene queso y tomate el nombre debe comenzar con **Pizza Clásica**
 - En caso contrario, debe solo comenzar con **Pizza**
- Luego, seguido del prefijo, debe venir el cuerpo del nombre:
 - Si tiene queso, tomate y aceituna o jamón, el cuerpo debe ser **Napolitana**
 - Si tiene jamón y pepperoni, el cuerpo debe ser **Carnívora**
 - Si tiene queso, tomate y champiñones o tomate, champiñones y aceitunas, el cuerpo debe ser **Vegetariana**
 - Si no cumple con ninguna de las anteriores, el cuerpo debe ser **Especial de la Casa**
- Finalmente, el nombre debe terminar con alguno de estos sufijos:
 - Si tiene 2 quesos, **Doble Queso con <Ingrediente restante>**
 - Si tiene 3 quesos, **Triple Queso**
 - Si tiene otro ingrediente (no queso) repetido 2 veces, debe decir **2x<Nombre Ingrediente>** con **<Nombre del ingrediente restante>**

- Si tiene otro ingrediente repetido 3 veces, debe decir **Big Special <Nombre Ingrediente> Deluxe**
- Si no cumple ninguna de las anteriores, debe listar los ingredientes comenzando con un guion y separando los ingredientes por coma: - <Ingrediente 1>, <Ingrediente 2>, <Ingrediente 3>

El nombre de los ingredientes puede ser dicho en cualquier orden. Si se pide un ingrediente que no existe, mostrar **error**.

Entrada	Salida
1 1 1	Ingresar ingredientes: Pizza Especial de la Casa Triple Queso
1 2 3	Ingresar ingredientes: Pizza Clásica Napolitana - Queso, Tomate, Aceituna
1 2 4	Ingresar ingredientes: Pizza Clásica Napolitana - Queso, Tomate, Jamón
1 4 5	Ingresar ingredientes: Pizza Carnívora - Queso, Jamón, Pepperoni
4 4 5	Ingresar ingredientes: Pizza Carnívora 2xJamón con Pepperoni
2 3 6	Ingresar ingredientes: Pizza Vegetariana - Tomate, Aceituna, Champiñones
5 5 5	Ingresar ingredientes: Pizza Big Special Pepperoni Deluxe
1 2 7	Ingresar ingredientes: error