**Ejercicio 4)**

**Ejercicio 5)**

=2,22

**Ejercicio 6)**

**Ejercicio 7):**

**Ejercicio 8):**

**Ejercicio 9):**

Es igual a:

**Ejercicio 10):**

Es igual a:

**Ejercicio 11):**

Es igual a:

**Ejercicio 12):**

**Análisis:** Se le solicita al usuario un nombre, para recibir un saludo personalizado por sistema.

**Diseño:**

1. Solicitar al usuario ingresar su nombre.
2. El sistema recibe el nombre ingresado por el usuario.
3. El sistema presenta en pantalla un saludo personalizado con el nombre ingresado.

**Ejercicio 13):**

**Análisis:** Se le solicita al usuario detallar base y altura de un rectángulo. El sistema analizará los valores y detallará área y perímetro.

**Diseño:**

1. Solicitar al usuario ingresar base y altura.
2. El sistema recibe los datos y realiza los cálculos.
3. El sistema presenta en pantalla área y perímetro.

**Ejercicio 14):**

**Análisis:** Se le solicita al usuario detallar cateto opuesto y cateto adyacente de un triángulo rectángulo. Sistema recibe los datos, procesa y devuelve el valor de la hipotenusa en pantalla.

**Diseño:**

1. Solicitar al usuario ingresar cateto opuesto y cateto adyacente.
2. El sistema recibe los datos y realiza los cálculos.
3. El sistema presenta en pantalla valor de la hipotenusa

**Ejercicio 15):**

**Análisis:** Sistema debe tomar dos números ingresados por el usuario y:

* Realizar la suma de los dos números.
* Realizar la resta del segundo número al primero.
* Realizar la multiplicación de los dos números.
* Realizar la división del primer número por el segundo.

**Diseño:** Usuario ingresa dos números. Sistema recibe valores y calcula la suma, resta, multiplicación y división de ellos. Al finalizar, muestra los resultados en pantalla.

**Ejercicio 16):**

**Análisis:** El usuario ingresa valores en grados Fahrenheit y el sistema le devuelve la conversión en grados Celsius.

**Diseño:** Usuario ingresa un valor numérico que representa grados Fahrenheit, el sistema recibe el dato, realiza la conversión y devuelve en pantalla el valor resultante.

**Ejercicio 17):**

**Análisis:** El personaje “Link” debe agarrar el premio que aparece en cualquier lugar posible de la pantalla. Cuando lo logra, el obsequio aparece en otra ubicación aleatoria para que Link proceda a agarrarlo nuevamente.

**Diseño:**

* Se define la posición inicial de Link y luego se define una posición aleatoria para el premio.
* Se calcula la distancia de Link con respecto al premio, sea cual fuese la ubicación de este último.
* Se establece la condición de que, si Link se acerca a distancias pequeñas del premio, éste aparece en otra ubicación aleatoria.
* Se modifica la ubicación de Link inicial por la que defina el puntero del mouse.

**Ejercicio 18):**

**Análisis:** Se le solicita al usuario detallar los valores de “a”, “b” y “c”. El sistema analizará los valores y detallará las dos raíces resultantes luego de calcular usando la fórmula cuadrática.

**Diseño:**

1. Solicitar al usuario ingresar valores a. b y c para una fórmula cuadrática.
2. El sistema recibe los datos y realiza los cálculos.
3. El sistema presenta en pantalla las dos raíces resultantes.

**Ejercicio 19):**

**Análisis:** En pantalla debe mostrarse una línea y una elipse. Esta última, ubicada en el medio de la elipse, se debe mover en conjunto con la línea desde el borde superior del lienzo hasta el borde inferior. Este movimiento no debe parar nunca.

**Diseño:**

1. Se deben declarar las variables necesarias correspondientes a las coordenadas de la línea y, por otro lado, una variable que será utilizada para la dirección de la misma.
2. Asignar valores a las variables de coordenadas para que se dibuje la línea en pantalla. En paralelo, dibujar la elipse en el punto medio exacto de la línea, ubicada a 40px de la línea en dirección vertical.
3. Para mover la línea, se debe especificar en casa secuencia la actualización de las variables de coordenadas de la línea. Programar para que se mueva de arriba hacia abajo.
4. A través de la variable de dirección se debe limitar el movimiento con el fin de que cuando alcance los limites del lienzo, cambie su dirección.

**Ejercicio 20):**

**Análisis:** En pantalla debe mostrarse un lienzo de un color determinado con rectángulos idénticos (de 440x420px), en color y tamaño (de 40x20px), que mantienen una misma distancia entre ellos (20px). Se debe utilizar “for” para generar la repetición de rectángulos.

**Diseño:**

1. Utilizar “for” dos veces: una para líneas y otra para columnas.
2. Cada vez que se itere, se debe calcular las coordenadas “x”, “y” de la esquina superior izquierda de cada rectángulo.
3. En cada cálculo de coordenada, se dibujará un rectángulo de 40x20px.

**Ejercicio 21):**

**Análisis:** En pantalla debe mostrarse una escalera que vaya desde el borde superior izquierdo del lienzo hasta el borde inferior derecho del mismo, con 8 escalones y un punto rojo ubicado en cada esquina superior derecha de cada escalón.

**Diseño:**

1. Se tiene que definir las variables correspondientes a la posición inicial de la escalera, las dimensiones de cada escalón y el número total de escalones.
2. En cada iteración se dibuja cada escalón utilizando solamente dos líneas, de modo tal que forme una letra “L” que de forma al escalón.
3. Se dibuja una elipse en cada esquina superior derecha de cada escalón.
4. Se actualizan las variables de posición de la escalera para moverla hacia abajo y después a la derecha.

**Ejercicio 22):**

**Análisis:** En pantalla deben mostrarse 6 renglones formados colocando 5 líneas en su interior y utilizando los bordes superior e inferior del lienzo para conformar el primer y el sexto renglón. En los renglones 1, 3 y 5 debe haber 5 círculos equidistantes entre sí, colocados sobre la línea inferior del renglón y deben tener colores aleatorios.

**Diseño:**

1. Se tiene que definir las variables correspondientes al número de líneas, espacio entre líneas, un contador para líneas y círculos y una que establezca la coordenada para “y” para dibujar estos elementos.
2. Se usa “do-while” para asegurar que se dibuje al menos una línea en el lienzo, y en cada iteración se verifica si la línea es impar para dibujar los círculos sobre ella. Finalmente se actualiza la coordenada de “y”.