

Problema I

```
while (true) {  
    var enemy1 = readline();  
    var dist1 = parseInt(readline());  
    var enemy2 = readline();  
    var dist2 = parseInt(readline());  
  
    if (dist1 < dist2) {  
        console.log(enemy1);  
    } else {  
        console.log(enemy2);  
    }  
}
```

¿Qué hace este código?

Bucle Infinito (while (true)):

El programa sigue funcionando sin parar, repitiendo las mismas acciones una y otra vez.

Leer Datos:

readline() pide al usuario que ingrese algo y lo guarda en una variable.

parseInt(readline()) convierte lo que el usuario ingresa en un número.

El programa pide el nombre y la distancia de dos enemigos:

enemy1 es el nombre del primer enemigo.

dist1 es la distancia del primer enemigo.

enemy2 es el nombre del segundo enemigo.

dist2 es la distancia del segundo enemigo.

Comparar Distancias:

El programa compara las dos distancias.

Si la distancia del primer enemigo (dist1) es menor, imprime el nombre del primer enemigo (enemy1).

Si no, imprime el nombre del segundo enemigo (enemy2).

Ejemplo

Si ingresas:

"EnemyA" y 10 para el primer enemigo.

"EnemyB" y 20 para el segundo enemigo.

El programa imprimirá "EnemyA" porque 10 es menor que 20.

En resumen, el programa siempre dice cuál de los dos enemigos está más cerca.

Problema II

```
while (true) {  
  let enemy1 = prompt("enemy1: ");  
  let dist1 = parseInt(prompt("dist1: "));  
  let enemy2 = prompt("enemy2: ");  
  let dist2 = parseInt(prompt("dist2: "));  
  
  if (dist1 < dist2) {  
    console.log(enemy1);  
  } else if (dist2 < dist1) {  
    console.log(enemy2);  
  } else {  
    console.log(enemy1); // En caso de que ambos esten a la misma distancia, se  
selecciona el primero  
  }  
}
```

¿Qué hace este código?

Bucle Infinito (while (true)):

El programa sigue funcionando sin parar, repitiendo las mismas acciones una y otra vez.

Leer Datos:

prompt("enemy1: ") pide al usuario que ingrese el nombre del primer enemigo y lo guarda en la variable enemy1.

parseInt(prompt("dist1: ")) pide al usuario que ingrese la distancia del primer enemigo, la convierte en un número y la guarda en la variable dist1.

prompt("enemy2: ") pide al usuario que ingrese el nombre del segundo enemigo y lo guarda en la variable enemy2.

parseInt(prompt("dist2: ")) pide al usuario que ingrese la distancia del segundo enemigo, la convierte en un número y la guarda en la variable dist2.

Comparar Distancias:

El programa compara las dos distancias.

Si la distancia del primer enemigo (dist1) es menor, imprime el nombre del primer enemigo (enemy1).

Si la distancia del segundo enemigo (dist2) es menor, imprime el nombre del segundo enemigo (enemy2).

Si las dos distancias son iguales, imprime el nombre del primer enemigo (enemy1).

En resumen, el programa siempre pregunta por dos enemigos y sus distancias, y luego dice cuál de los dos está más cerca.

Problema III

```
while (true) {  
  let enemy1 = prompt("enemy1: ");  
  let dist1 = parseInt(prompt("dist1: "));  
  let enemy2 = prompt("enemy2: ");  
  let dist2 = parseInt(prompt("dist2: "));  
  
  if (dist1 < dist2) {  
    console.log(enemy1);  
  } else {  
    console.log(enemy2);  
  }  
}
```

¿Qué hace este código?

Bucle Infinito (while (true)):

El programa repite las mismas acciones una y otra vez sin parar, lo que permite manejar entradas que cambian rápidamente.

Leer Datos:

prompt("enemy1: ") pide al usuario que ingrese el nombre del primer enemigo y lo guarda en la variable enemy1.

parseInt(prompt("dist1: ")) pide al usuario que ingrese la distancia del primer enemigo, la convierte en un número y la guarda en la variable dist1.

prompt("enemy2: ") pide al usuario que ingrese el nombre del segundo enemigo y lo guarda en la variable enemy2.

parseInt(prompt("dist2: ")) pide al usuario que ingrese la distancia del segundo enemigo, la convierte en un número y la guarda en la variable dist2.

Comparar Distancias:

Si la distancia del primer enemigo (dist1) es menor, imprime el nombre del primer enemigo (enemy1).

De lo contrario, imprime el nombre del segundo enemigo (enemy2).

Ejemplo.

Supongamos que ingresas los siguientes datos en la primera iteración:

"Enemy1" y 50 para el primer enemigo.

"Enemy2" y 30 para el segundo enemigo.

El programa imprimirá "Enemy2" porque 30 es menor que 50.

En la siguiente iteración, si ingresas:

"Enemy1" y 25 para el primer enemigo.

"Enemy2" y 45 para el segundo enemigo.

El programa imprimirá "Enemy1" porque 25 es menor que 45.

Resumen.

Este código constantemente pregunta por los nombres y distancias de dos enemigos y luego determina cuál está más cerca, asegurando que maneje correctamente entradas que cambian rápidamente.

Problema IV

```
while (true) {  
  let enemy1 = prompt("enemy1: ");  
  let dist1 = parseInt(prompt("dist1: "));  
  let priority1 = parseInt(prompt("priority1: "));  
  let enemy2 = prompt("enemy2: ");  
  let dist2 = parseInt(prompt("dist2: "));  
  let priority2 = parseInt(prompt("priority2: "));  
  
  if (priority1 > priority2) {  
    console.log(enemy1);  
  } else if (priority2 > priority1) {  
    console.log(enemy2);  
  } else {  
    if (dist1 < dist2) {  
      console.log(enemy1);  
    } else {  
      console.log(enemy2);  
    }  
  }  
}
```

¿Qué hace este código?

Bucle Infinito (while (true)):

El programa sigue funcionando sin parar, repitiendo las mismas acciones una y otra vez.

Leer Datos:

prompt("enemy1: ") pide al usuario que ingrese el nombre del primer enemigo y lo guarda en la variable enemy1.

parseInt(prompt("dist1: ")) pide al usuario que ingrese la distancia del primer enemigo, la convierte en un número y la guarda en la variable dist1.

parseInt(prompt("priority1: ")) pide al usuario que ingrese la prioridad del primer enemigo y la guarda en la variable priority1.

prompt("enemy2: ") pide al usuario que ingrese el nombre del segundo enemigo y lo guarda en la variable enemy2.

parseInt(prompt("dist2: ")) pide al usuario que ingrese la distancia del segundo enemigo, la convierte en un número y la guarda en la variable dist2.

parseInt(prompt("priority2: ")) pide al usuario que ingrese la prioridad del segundo enemigo y la guarda en la variable priority2.

Comparar Prioridades y Distancias:

El programa compara las prioridades de los dos enemigos.

Si la prioridad del primer enemigo (priority1) es mayor, imprime el nombre del primer enemigo (enemy1).

Si la prioridad del segundo enemigo (priority2) es mayor, imprime el nombre del segundo enemigo (enemy2).

Si ambas prioridades son iguales, compara las distancias:

Si la distancia del primer enemigo (dist1) es menor, imprime el nombre del primer enemigo (enemy1).

De lo contrario, imprime el nombre del segundo enemigo (enemy2)

.

Ejemplo

Supongamos que ingresas los siguientes datos:

Primera ronda de datos:

enemy1: "EnemyA"

dist1: 20

priority1: 2

enemy2: "EnemyB"

dist2: 15

priority2: 3

El programa imprimirá "EnemyB" porque su prioridad (3) es mayor que la prioridad de "EnemyA" (2).

Segunda ronda de datos:

enemy1: "EnemyA"

dist1: 25

priority1: 2

enemy2: "EnemyB"

dist2: 25

priority2: 2

Las prioridades son iguales, así que el programa comparará las distancias. Imprimirá "EnemyA" porque las distancias son iguales y el primer enemigo se selecciona en caso de empate.

Resumen

Objetivo: Determinar cuál enemigo atacar basado en su prioridad y distancia.

Pasos:

Pedir nombre, distancia, y prioridad de dos enemigos.

Comparar las prioridades:

Imprimir el nombre del enemigo con mayor prioridad.

Si las prioridades son iguales, comparar las distancias:

Imprimir el nombre del enemigo con menor distancia.

Continuar el ciclo de forma indefinida, esperando nuevas entradas.

Este código es útil para decidir qué enemigo atacar primero en base a su prioridad y distancia.

Problema V

```
while (true) {  
  let enemy1 = prompt("enemy1: ");  
  let dist1 = parseInt(prompt("dist1: "));  
  let priority1 = parseInt(prompt("priority1: "));  
  let speed1 = parseInt(prompt("speed1: "));  
  let enemy2 = prompt("enemy2: ");  
  let dist2 = parseInt(prompt("dist2: "));  
  let priority2 = parseInt(prompt("priority2: "));  
  let speed2 = parseInt(prompt("speed2: "));  
  
  if (priority1 > priority2) {  
    console.log(enemy1);  
  } else if (priority2 > priority1) {  
    console.log(enemy2);  
  } else {  
    if (dist1 < dist2) {  
      console.log(enemy1);  
    } else if (dist2 < dist1) {  
      console.log(enemy2);  
    } else {  
      if (speed1 > speed2) {  
        console.log(enemy1);  
      } else {  
        console.log(enemy2);  
      }  
    }  
  }  
}
```

¿Qué hace este código?

Este código está diseñado para decidir cuál enemigo atacar en función de su prioridad, distancia y velocidad.

Bucle Infinito (while (true)):

El programa sigue funcionando sin parar, repitiendo las mismas acciones una y otra vez.

Leer Datos:

prompt("enemy1: ") pide al usuario que ingrese el nombre del primer enemigo y lo guarda en la variable enemy1.

parseInt(prompt("dist1: ")) pide al usuario que ingrese la distancia del primer enemigo, la convierte en un número y la guarda en la variable dist1.

`parseInt(prompt("priority1: "))` pide al usuario que ingrese la prioridad del primer enemigo y la guarda en la variable `priority1`.

`parseInt(prompt("speed1: "))` pide al usuario que ingrese la velocidad del primer enemigo y la guarda en la variable `speed1`.

`prompt("enemy2: ")` pide al usuario que ingrese el nombre del segundo enemigo y lo guarda en la variable `enemy2`.

`parseInt(prompt("dist2: "))` pide al usuario que ingrese la distancia del segundo enemigo, la convierte en un número y la guarda en la variable `dist2`.

`parseInt(prompt("priority2: "))` pide al usuario que ingrese la prioridad del segundo enemigo y la guarda en la variable `priority2`.

`parseInt(prompt("speed2: "))` pide al usuario que ingrese la velocidad del segundo enemigo y la guarda en la variable `speed2`.

Comparar Prioridades, Distancias y Velocidades:

Comparar Prioridades:

Si la prioridad del primer enemigo (`priority1`) es mayor, imprime el nombre del primer enemigo (`enemy1`).

Si la prioridad del segundo enemigo (`priority2`) es mayor, imprime el nombre del segundo enemigo (`enemy2`).

Si las Prioridades son Iguales:

Compara las distancias:

Si la distancia del primer enemigo (`dist1`) es menor, imprime el nombre del primer enemigo (`enemy1`).

Si la distancia del segundo enemigo (`dist2`) es menor, imprime el nombre del segundo enemigo (`enemy2`).

Si las distancias son iguales, compara las velocidades:

Si la velocidad del primer enemigo (`speed1`) es mayor, imprime el nombre del primer enemigo (`enemy1`).

De lo contrario, imprime el nombre del segundo enemigo (`enemy2`).

Ejemplo

Supongamos que ingresas los siguientes datos:

Primera ronda de datos:

```
enemy1: "EnemyA"  
dist1: 30  
priority1: 2  
speed1: 10  
enemy2: "EnemyB"  
dist2: 30  
priority2: 2  
speed2: 20
```

Las prioridades son iguales, las distancias son iguales, así que el programa compara las velocidades. Imprimirá "EnemyB" porque su velocidad (20) es mayor que la velocidad de "EnemyA" (10).

Segunda ronda de datos:

```
enemy1: "EnemyA"  
dist1: 25  
priority1: 2  
speed1: 10  
enemy2: "EnemyB"  
dist2: 45  
priority2: 2  
speed2: 20
```

Las prioridades son iguales, así que el programa compara las distancias. Imprimirá "EnemyA" porque su distancia (25) es menor que la distancia de "EnemyB" (45).

Resumen

Objetivo: Determinar cuál enemigo atacar basado en su prioridad, distancia y velocidad.

Pasos:

Pedir nombre, distancia, prioridad y velocidad de dos enemigos.

Comparar las prioridades:

Imprimir el nombre del enemigo con mayor prioridad.

Si las prioridades son iguales, comparar las distancias:

Imprimir el nombre del enemigo con menor distancia.

Si las distancias son iguales, comparar las velocidades:

Imprimir el nombre del enemigo con mayor velocidad.

Continuar el ciclo de forma indefinida, esperando nuevas entradas. Este código es útil para decidir qué enemigo atacar primero en base a su prioridad, distancia y velocidad.