Lab 3 [Battle Scene]- Logics

In this workshop, we will continue our development of a text based game "Code Quest!". This workshop focuses on retrieving input from the user, selecting from multiple logic paths based on user input and looping until some kind of condition is met.

LEARNING OUTCOMES

Upon successful completion of this workshop, you will be able to

- Write menus to interact with the user
- Use user input to determine which conditional path to use
- Use loops to repeat a set of operations until a goal is complete

PART 1:

SPECIFICATIONS

In this lab you are going to create a program `w3.c`, you are going to simulate a battle between a player and an enemy.

- Create player stat variables for `Strength`, `Defense`, `Intelligence`, and `HP` and initialize them as 10, 20, 40, and 10 respectively.
- Create enemy stat variables for `Strength`, `Defense`, `Intelligence`, and `HP` and initialize them as 30, 40, 25, and 10 respectively.
- Display "**Battle Start!**" and begin the battle sequence (below).
- Display to the user the current `HP` of the player and the enemy.
- Display a menu to the user with the attack options and accept the input from user (either '1' or '2')
- Option 1: If the user selects '1' (attack):
  - `attack power`: Calculate the attack power by dividing the `strength` of the player by the `defense` of the enemy
  - Multiply the calculated `attack power` by 5 and reduce the enemy's `HP` by that amount, truncating `HP` to a whole number
  - e.g.: enemy HP (10) - player attack (10) / enemy defence (40) * 5 = 8.75 → 8
- Option 2: If the user selects '2' (magic):
  - `magic power`: Calculate the magic power by dividing the `Intelligence` of the player by the `Intelligence` of the enemy
  - Multiply the calculated `magic power` by 5 and reduce the enemy's `HP` by that amount, truncating `HP` to a whole number
- If the enemy's `HP` is zero or less, indicate to the user that they won, if not the enemy attacks the player (follow the attack process above, swap the enemy and player).
- If the player's `HP` is zero or less, indicate to the user that they have lost.

- Repeat the above battle sequence until one either the player or the enemy have run out of `HP.`
- Once the battle has been decided, the program exits.

The output of two sample runs look like this (user input highlighted in green):

You may assume that the user enters valid numbers. If your program's output exactly matches the output shown above, given the provided inputs, then your lab is complete and ready to be submitted (read below).

Sample 1:

```
Battle Start!
Your HP: 10 Enemy HP: 10
1 - Attack
2 - Magic
Select your move: 1
You attacked the enemy!
The enemy attacked you!

Your HP: 2 Enemy HP: 8
1 - Attack
2 - Magic
Select your move: 1
You attacked the enemy!
The enemy attacked you!
You died!
```

Sample 2:

```
Battle Start!
Your HP: 10 Enemy HP: 10
1 - Attack
2 - Magic
Select your move: 2
You shocked the enemy!
The enemy attacked you!
Your HP: 2 Enemy HP: 2
1 - Attack
2 - Magic
Select your move: 2
You shocked the enemy!
You Won!
```

PART 2 (BONUS):

SPECIFICATIONS

BUILDING THE GAME

If you have completed all previous game labs, you may complete this section in order to merge your current lab with you previous labs.

Copy the source code from this lab (excluding the main and braces) and open up your "w2.c" file. Paste the copied source code into the code quest source file below the player stats section within the main. Remove the variable declarations for the player stats (except HP) and change this lab code to use the player stat variables from lab 2. If successful, the output of the program should have the intro screen and followed by the character creation section and the battles should commence differently depending on the stats chosen in the player creation section.

Prepare a Typescript

Create a typescript on the remote Linux host using the following commands:

```
+ At the prompt, type: script w3.txt
+ At the prompt, type: whoami
+ At the prompt, type: cat w3.c
+ At the prompt, type: gcc w3.c -o w3.out
+ At the prompt, type: w3.out
+ At the prompt type: exit
```

This will produce a typescript named **"w3.txt"** in your current directory. Transfer a copy of this file to your local computer.

SUBMISSION

Upload your typescript file to BlackBoard:

- Login to BlackBoard
- Select your course code
- Select Workshop 3 under Workshops
- Upload `w3.txt`
- Write a short note to your instructor
  - Under “Add comments”, add a sentence or two regarding what you think you learned in this workshop in the notes textbox
  - press "Save Changes"
- When ready to submit, press "Submit". Note you can save a draft until you are ready to submit.