# CSE 102 Spring 2025 – Computer Programming Assignment 11

## Due on May 28, 2025 at 23:59

Write a C program to implement Wizard War. In a mystical arena, two powerful wizards engage in a duel using elemental spells. They take turns casting spells or meditating to recover mana. The battle continues recursively until one wizard is defeated. After the duel, each wizard's **performance score** is calculated using recursion based on their actions.

Each wizard includes:

- Name: string
- Magic class: string (e.g., fire, ice)
- Three spells: randomly selected at the start
- Health Points (HP): starts at 100
- Mana: starts at 100
- Alive: boolean (1 = alive, 0 = dead)
- Battle Stats: must track during duel
    - Total damage dealt
    - Total mana spent
    - Number of recoveries (meditations)

Each spell has:

- Name: string
- Magic class: string
- Minimum damage, Maximum damage: integers
- Minimum mana cost, Maximum mana cost: integers

All spells are loaded from spellbook.txt. Each line contains:

name,class,min_damage,max_damage,min_mana,max_mana

Example:

Fireball,fire,10,25,10,15

Inferno,fire,15,30,15,20

Flame Surge,fire,12,22,12,18

Ice Shard,ice,8,20,9,14

Freeze,ice,10,24,11,16

Blizzard,ice,14,28,14,20

You must define and use the following functions:

void duel(Wizard* attacker, Wizard* defender);

When each wizard is defined, their spells are selected from this book randomly without regarding the same magic class.

Use recursion for turn-based duel logic:

void duel(Wizard* attacker, Wizard* defender);

**Game Flow:**

1. Attacker randomly chooses one of their 3 spells.
2. Calculate a **random mana cost** and **random damage** using that spell's range.
3. If the attacker's mana is **insufficient**:
    - They **recover** a random amount of mana (10–20).
    - Track this recovery in their stats.
4. If the caster has enough mana:
    - Apply the damage to the defender.
    - If **magic class matches** the spell's class, add **+5 bonus damage**.
    - Deduct mana from the caster.
    - Update caster's **damage dealt** and **mana spent** stats.
5. If the defender's HP drops to 0 or below, the attacker wins.
6. Otherwise, recursively continue the duel with roles swapped.

After the duel, implement a recursive function to compute each wizard's **performance score**:

int calculate_score(int stats[], int n);

Where:

- stats[0] = total damage dealt
- stats[1] = total mana spent
- stats[2] = number of recoveries
- n = 3

**Formula: score = damage × 2 + mana spent × 1 – recoveries × 3**

**EXAMPLE OUTPUT**

Wizard Duel Begins: Merlin vs Frostina!

---

Merlin casts Fireball on Frostina!

Damage: 18 | Frostina's HP: 82 | Merlin's Mana: 90

---

Frostina casts Ice Shard on Merlin!

Damage: 15 | Merlin's HP: 85 | Frostina's Mana: 90

...

Merlin is low on mana and meditates...

Merlin recovers 18 mana. Current mana: 33

...

Frostina casts Blizzard on Merlin!

Damage: 25 | Merlin's HP: 0 | Frostina's Mana: 60

---

Winner: Frostina the Ice Wizard!

Battle Summary:

Merlin — Damage: 72 | Mana Spent: 88 | Recoveries: 2 | Score: 242

Frostina — Damage: 100 | Mana Spent: 95 | Recoveries: 1 | Score: 289

**Requirements:**

- Use struct for both Wizard and Spell.
- Use **recursion** for:
  - duel()
  - calculate_score()
- Use **rand()** for:
  - Spell selection
  - Damage and mana cost ranges
  - Recovery values (10–20)
- Load spell data from spellbook.txt
- Output must follow the given format.
- Wizard configuration should be **hardcoded in main()**.
- Each wizard should get **3 randomly assigned spells** (they may overlap).
- No global arrays, loops, or pointer arithmetic in recursive logic.

---

IMPORTANT NOTES:

- Submit your homework as a zip file named as your name_surname id (name_surname.zip) and this file should include:
  - name_surname.c file
- name_surname.pdf file which includes, screenshots of your generated outputs and given C code as an input.
- Programs with compilation errors will get 0.
- The output format must be as given, do not change it.
- Compile your work with given command “gcc --ansi your_program.c -o your_program”.
- Your work will be evaluated using gcc version 11.4.0.
- For any questions and problems, you can always contact with me **via e-mail** (gizemsungu@gtu.edu.tr) or you can find me in Room 234 in scheduled office hours in May 14 and May 28, 2025 between 13:30 – 14:30.