CSE 102
ASSIGNMENT 11
MUHAMMET FATİH
ALBAYIN

THE CODE

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>

typedef struct {                    /* To keep wizard data
*/
    char name[20];
    char magic_class[20];
    int min_dmg;
    int max_dmg;
    int min_mana_cost;
    int max_mana_cost;
}spell_type;

typedef struct {                    /* To keep spell data
*/
    char name[20];
    char magic_class[20];
    spell_type spell1;
    spell_type spell2;
    spell_type spell3;
    int hp;
    int mana;
    int total_damage_dealt;
    int total_mana_spent;
    int recovery_count;
    bool alive;
}wizard_type;


void read_spells(spell_type spells[6]);
void duel(wizard_type *wizard1, wizard_type *wizard2);
int calculate_score(int stats[], int n);
void randomize_spells(wizard_type *wizard, spell_type spells[], int
spell_amount);
spell_type choose_spell(wizard_type *wizard);
int random_damage(spell_type spell);
int random_mana(spell_type spell);
void initialize_wizard(wizard_type *wizard);


int main() {
    spell_type spells[6];           /* Array to keep all the spells
*/
    int spell_amount = 6, i;
    srand(time(NULL));

    read_spells(spells);            /* Spells are read from the file
*/

```

```c
    wizard_type hokkabaz, kahin;    /* Wizards are named
*/
    strcpy(hokkabaz.name, "Hokkabaz");
    strcpy(kahin.name, "Kahin");

    randomize_spells(&hokkabaz, spells, spell_amount);      /* Spells are
randomly assigned             */
    randomize_spells(&kahin, spells, spell_amount);

    initialize_wizard(&hokkabaz);                           /* Some wizard
vairables are initialized    */
    initialize_wizard(&kahin);

    printf("Wizard Duel Begins: %s vs %s!\n", hokkabaz.name, kahin.name);
    duel(&hokkabaz, &kahin);                                /* Duel progress
*/

    if(hokkabaz.alive == 1)                                 /* Winner is printed
*/
        printf("\nWinner: %s the %s Wizard!\n", hokkabaz.name,
hokkabaz.magic_class);
    else
        printf("\nWinner: %s the %s Wizard!\n", kahin.name, kahin.magic_class);

    int stats[3];                                           /* Array to keep
stats of the wizard        */
    stats[0] = hokkabaz.total_damage_dealt;
    stats[1] = hokkabaz.total_mana_spent;
    stats[2] = hokkabaz.recovery_count;

    int score_h = calculate_score(stats, 3);                /* Scores for each
wizard is calculated     */

    stats[0] = kahin.total_damage_dealt;
    stats[1] = kahin.total_mana_spent;
    stats[2] = kahin.recovery_count;

    int score_k = calculate_score(stats, 3);

    printf("\nBattle Summary:\n");
    printf("%s — Damage: %d | Mana Spent: %d | Recoveries: %d | Score: %d\n",
hokkabaz.name, hokkabaz.total_damage_dealt, hokkabaz.total_mana_spent,
hokkabaz.recovery_count, score_h);
    printf("%s — Damage: %d | Mana Spent: %d | Recoveries: %d | Score: %d\n",
kahin.name, kahin.total_damage_dealt, kahin.total_mana_spent,
kahin.recovery_count, score_k);

    return 0;
}
```

```c
void read_spells(spell_type spells[6]) {        /* Spell data is read and stored in the array       */
    FILE* fptr = fopen("spelldata.txt", "r");

    char line[50];
    int i = 0, j;
    while(i < 6 && fgets(line, sizeof(line), fptr)) {
        line[strcspn(line,"\n")] = 0;
        char *token = strtok(line, ",");
        for(j = 0; j < 6 && token != NULL; j++) {
            if(j == 0) strcpy(spells[i].name, token);
            else if(j == 1) strcpy(spells[i].magic_class, token);
            else if(j == 2) spells[i].min_dmg = atoi(token);
            else if(j == 3) spells[i].max_dmg = atoi(token);
            else if(j == 4) spells[i].min_mana_cost = atoi(token);
            else if(j == 5) spells[i].max_mana_cost = atoi(token);

            token = strtok(NULL, ",");
        }
        i++;
    }
}


void duel(wizard_type *attacker, wizard_type *defender) {   /* Duel process is done recursively     */
    int rnd, damage, mana_cost, recovery;

    if(attacker->hp <= 0 || defender->hp <= 0)
        return;


    spell_type spell = choose_spell(attacker);

    damage = random_damage(spell);              /* Damage and mana values are randomized        */
    mana_cost = random_mana(spell);

    if(mana_cost > attacker->mana) {            /* Player recovers mana if needed               */
        recovery = (rand() % 10) + 10;

        attacker->mana += recovery;

        printf("-------------------------------------------------------\n");
        printf("%s is low on mana and meditates ...\n",attacker->name);
        printf("%s recovers %d mana. Current mana: %d\n", attacker->name, recovery, attacker->mana);
        attacker->recovery_count += 1;
        duel(defender, attacker);
    }

```

```
133     else {
134 
135     if(strcmp(spell.magic_class, attacker->magic_class) == 0)
136         damage += 5;                /* Extra damage is added if types match
    */
137 
138     defender->hp -= damage;         /* Wizard variables are updated
    */
139     attacker->mana -= mana_cost;
140     attacker->total_damage_dealt += damage;
141     attacker->total_mana_spent += mana_cost;
142 
143     if(defender->hp <= 0)
144         defender->alive = 0;
145 
146     printf("-------------------------------------------------------\n");
147     printf("%s casts %s on %s!\n", attacker->name, spell.name, defender->name);
148     printf("Damage: %d | %s's HP: %d | %s's Mana: %d\n", damage, defender->name,
    defender->hp, attacker->name, attacker->mana);
149 
150     duel(defender, attacker);
151     }
152 }
153 
154 
155 int calculate_score(int stats[], int n) {
156     if(n == 0) return 0;            /* Score is calculated recursively
    */
157 
158     int value = 0;
159     if(n == 3)
160         value = stats[2] * (-3);
161     else if(n == 2)
162         value = stats[1];
163     else if(n == 1)
164         value = stats[0] * 2;
165 
166     return value + calculate_score(stats, n-1);
167 }
168 
169 
170 void randomize_spells(wizard_type *wizard, spell_type spells[], int
    spell_amount) {
171 
172     wizard->spell1 = spells[rand() % 6];        /* All three spells are randomly
    assigned to the wizard and magic class is determined   */
173     wizard->spell2 = spells[rand() % 6];
174     wizard->spell3 = spells[rand() % 6];
175 
176     strcpy(wizard->magic_class, spells[rand() % 6].magic_class);
177 
178 }
```

```
179
180
181 spell_type choose_spell(wizard_type *wizard) {
182     int rnd = rand() % 3;                          /* Spell is randomly chosen for
    each move       */
183
184     if(rnd == 0) return wizard->spell1;
185     else if(rnd == 1) return wizard->spell2;
186     else if(rnd == 2) return wizard->spell3;
187 }
188
189
190 int random_damage(spell_type spell) {
191     int damage;                                    /* Damage is randomized
    */
192
193     damage = (rand() % (spell.max_dmg - spell.min_dmg + 1)) + spell.min_dmg;
194
195     return damage;
196 }
197
198
199 int random_mana(spell_type spell) {
200     int mana_cost;                                 /* Mana cost is randomized
    */
201
202     mana_cost = (rand() % (spell.max_mana_cost - spell.min_mana_cost + 1)) +
    spell.min_mana_cost;
203
204     return mana_cost;
205 }
206
207
208 void initialize_wizard(wizard_type *wizard) {    /* Some variables of the wizard
    are initialized */
209     wizard->alive = 1;
210     wizard->hp = 100;
211     wizard->mana = 100;
212     wizard->recovery_count = 0;
213     wizard->total_damage_dealt = 0;
214     wizard->total_mana_spent = 0;
215 }
```

THE OUTPUT

```
albay@albay-VirtualBox:~/Desktop$ gcc -ansi 240104004064.c -o q
albay@albay-VirtualBox:~/Desktop$ ./q
Wizard Duel Begins: Hokkabaz vs Kahin!
-------------------------------------------------
Hokkabaz casts Ice Shard on Kahin!
Damage: 13 | Kahin's HP: 87 | Hokkabaz's Mana: 89
-------------------------------------------------
Kahin casts Inferno on Hokkabaz!
Damage: 22 | Hokkabaz's HP: 78 | Kahin's Mana: 82
-------------------------------------------------
Hokkabaz casts Blizzard on Kahin!
Damage: 22 | Kahin's HP: 65 | Hokkabaz's Mana: 69
-------------------------------------------------
Kahin casts Flame Surge on Hokkabaz!
Damage: 18 | Hokkabaz's HP: 60 | Kahin's Mana: 67
-------------------------------------------------
Hokkabaz casts Ice Shard on Kahin!
Damage: 18 | Kahin's HP: 47 | Hokkabaz's Mana: 55
-------------------------------------------------
Kahin casts Flame Surge on Hokkabaz!
Damage: 24 | Hokkabaz's HP: 36 | Kahin's Mana: 54
-------------------------------------------------
Hokkabaz casts Blizzard on Kahin!
Damage: 28 | Kahin's HP: 19 | Hokkabaz's Mana: 36
-------------------------------------------------
Kahin casts Inferno on Hokkabaz!
Damage: 22 | Hokkabaz's HP: 14 | Kahin's Mana: 39
-------------------------------------------------
Hokkabaz casts Ice Shard on Kahin!
Damage: 15 | Kahin's HP: 4 | Hokkabaz's Mana: 25
-------------------------------------------------
Kahin casts Fireball on Hokkabaz!
Damage: 25 | Hokkabaz's HP: -11 | Kahin's Mana: 26

Winner: Kahin the fire Wizard!

Battle Summary:
Hokkabaz - Damage: 96 | Mana Spent: 75 | Recoveries: 0 | Score: 267
Kahin - Damage: 111 | Mana Spent: 74 | Recoveries: 0 | Score: 296
albay@albay-VirtualBox:~/Desktop$
```

SPELL DATA FILE