

RPG

Pažljivo pročitati sekciju [Napomene](#).

Zadatak

Napisati program koji obrađuje podatke o moćima koje neki heroj može da ima u jednoj RPG igri. Podaci o moćima su zadati u ulaznoj datoteci u sledećem formatu:

- naziv (jedna reč, do 25 karaktera)
- tip (jedna reč, do 25 karaktera)
- cooldown (realan broj)
- damage (realan broj)

Učitati moći u jednostruko spregnutu listu, gde se novi čvorovi dodaju sortirani po DPS-u, i to opadajuće

DPS se računa na sledeći način: $DPS = damage/cooldown$

Program treba da sadrži implementacije sledećih funkcija (sa identičnim zaglavljima koja su ovde navedena, bez komentara koji opisuju šta funkcija radi):

```
// pokušava da otvori datoteku, izlazi iz programa u slučaju neuspešnog otvaranja
FILE *safe_fopen(char *naziv, char *rezim, int kod_greske);
// dinamički alokira memoriju, uz odgovarajuću proveru,
// popunjava cvor sa vrednostima parametara funkcije i vraća adresu na novi cvor
MOC *napravi_cvor(char *naziv, char *tip, float cooldown, float damage);
// učitava podatke iz ulazne datoteke u jednostruko spregnutu listu, red po red
void učitaj_moci(FILE *ulazna, MOC **pglava);
// ispisuje u fajl nazive i DPS svih moci koje deluju DPS iznad traženog
void ispis_adekvatnih(MOC *glava, float trazeni_dps, FILE *izlazna);
// ispisuje na standardni izlaz sve informacije o moci koja ima najveći DPS za taj tip heroja
void ispis_najbolje_moci(MOC *glava, char* tip);
// dodaje novi cvor u listu tako da lista bude sortirana po DPS u opadajućem redosledu
void dodaj_sortirano(MOC **pglava, MOC *novi);
// postavlja glavu liste na vrednost NULL
void inicijalizacija(MOC **pglava);
// dinamički oslobadja memoriju svakog cvora (koristiti valgrind za proveru) i glavu liste postavlja na NULL
void obrisi_listu(MOC **pglava);
```

Primer ulazne i očekivane izlazne datoteke biće dati na samoj odbrani. Sastaviti testnu ulaznu datoteku i testirati rešenje pre predaje. Prilikom predaje, očistiti main funkciju, tako da izgleda na sledeći način:

```
int main()
{

    return 0;
}
```

U slučaju uspešnog izvršavanja programa, izaći sa status kodom 0 (EXIT_SUCCESS). Ukoliko program ne može da se izvrši do kraja usled sledećih nedostataka, izaći iz programa sa sledećim status kodovima:

- U slučaju nedovoljnog ili suvišnog broj argumenata komandne linije, izaći iz programa sa status kodom 13.
- U slučaju nemogućnosti dinamičkog zauzimanja memorije, izaći iz programa sa status kodom 14.
- U slučaju loše otvorene ulazne datoteke, izaći iz programa sa status kodom 15.
- U slučaju loše otvorene izlazne datoteke, izaći iz programa sa status kodom 16.

Napomene

- Programski kod mora biti snimljen u okviru home direktorijuma, na predviđenom mestu
- Nije dozvoljeno snimanje više kopija ili varijacija izvornog koda, samo jedna .c datoteka
- Prilikom rada programa poštovati očekivani format teksta, prikazanom u tekstu zadatka
- Nije dozvoljeno koristiti globalne promenljive
- Programski kod ne sme sadržati sintaksne greške, niti upozorenja
- Obavezno osloboditi dinamički zauzetu memoriju
- Funkcije treba obavezno da poštuju zaglavlja zadata u zadatku
- Testirati sve detaljno pre predaje, ali rešenje predati sa praznom main funkcijom (bez parametara za argumente komandne linije, samo sa return 0 naredbom)
- Struktura se po potrebi može proširiti dodatnim poljima