

Računi za domaćinstvo

Pažljivo pročitati sekciju [Napomene](#).

Zadatak

Napisati program koji obrađuje podatke o prispelim računima. Podaci su zadati u ulaznoj datoteci u sledećem formatu:

- *naziv* (jedna reč, do 20 karaktera)
- *iznos* (realan broj)
- *rok_za_placanje* (celobrojna vrednost, interpretira se kao datum, na primer 25 označava 25. u mesecu)

Učitati reči u jednostruko spregnutu listu, gde se novi čvorovi dodaju sortirano po iznosu, u rastućem poretku.

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

```
// pokusava da otvori datoteku, izlazi iz programa u slucaju neuspešnog otvaranja
FILE *safe_fopen(char *naziv, char *rezim, int kod_greske);

// učitava podatke iz ulazne datoteke u jednostruko spregnutu listu, red po red
void ucitaj_racune(FILE *ulazna, RACUN **pglava);

// ispisuje racune ciji rok za placanje je datuma koji je prosledjen ili se kasni sa uplatom
// pre provere, na standardni izlaz ispisati tekst u formatu: "Pokusano hitno placanje racuna <naziv>: "
// ukoliko je moguće, ispisati informacije o imenu, iznosu, roku za placanje, kao i datumu kad je placen racun u izlaznu datoteku
// i "USPESNO" na standardni izlaz
// ako budžeta nema, ispisati "NEUSPESNO" na standardni izlaz
// funkcija vraća preostali budžet kao povratnu vrednost
double ispisi_hitne_racune(FILE *izlazna, RACUN *glava, int datum, double budzet);

// ispisuje racune ciji rok za placanje još uvek nije prispeo u izlaznu datoteku ukoliko postoji dovoljno budžeta da se oni plate
// ukoliko je moguće, ispisati informacije o imenu, iznosu, roku za placanje, kao i datumu kad je placen racun u izlaznu datoteku
// ako budžeta nema, ispisati na standardni izlaz tekst u formatu: "Racun <naziv> nije placen, rok je do <rok-za-placanje>."
// funkcija vraća preostali budžet kao povratnu vrednost
double ispisi_nadolazece_racune(FILE *izlazna, RACUN *glava, int datum, double budzet);

// postavlja glavu liste na vrednost NULL
void inicijalizacija(RACUN **pglava);

// dinamički alokira memoriju, uz odgovarajuću proveru,
// popunjava cvor sa vrednostima parametara funkcije i vraća adresu na novi cvor
RACUN *napravi_cvor(char *naziv, double iznos, int rok_za_placanje);

// dodaje novi cvor u listu tako da lista bude sortirana po iznosu u rastućem redosledu
void dodaj_sortirano(RACUN **pglava, RACUN *novi);

// dinamički oslobadja memoriju svakog cvora (koristiti valgrind za proveru) i glavu liste postavlja na NULL
void obrisi_listu(RACUN **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 1 (`EXIT_FAILURE`)
- U slučaju nemogućnosti dinamičkog zauzimanja memorije, izaći iz programa sa status kodom 2
- Ako program ne može da otvori ulaznu datoteku, izaći sa status kodom 3
- Ako program ne može da otvori izlaznu datoteku, izaći sa status kodom 4

Napomene

- Programski kod ne sme sadržati sintaksne greške, niti upozorenja
- 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
- 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)
- Podrazumevati (ne praviti dodatne validacije) da su svi datumi u programu (datumi ulaznih podataka, kao i argumenta komandne linije) brojevi između 1 i 30