

# Statistika država

Pažljivo pročitati sekciju [Napomene](#).

## Zadatak

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

- *naziv* (jedna reč, do 20 karaktera)
- *povrsina* (realan broj)
- *broj\_stanovnika* (celobrojna vrednost)
- *bdp* (realan broj)

Učitati reči u jednostruko spregnutu listu, gde se novi čvorovi dodaju sortirano po broju stanovnika, u opadajućem poretku.

Program treba da sadrži implementacije sledećih funkcija (sa identičnim zaglavljljima 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);

// ucitava podatke iz ulazne datoteke u jednostruko spregnutu listu, red po red
void ucitaj_drzave(FILE *ulazna, DRZAVA **pglava);

// ispisuje naziv, gustinu naseljenosti u BDP po stanovniku u izlaznu datoteku
void ispisi_statistiku(FILE *izlazna, DRZAVA *glava);

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

// dinamicki alocira memoriju, uz odgovarajucu proveru,
// popunjava cvor sa vrednostima parametara funkcije i vraca adresu na novi cvor
DRZAVA *napravi_cvor(char *drzava, double povrsina, int broj_stanovnika, double bdp);

// dodaje novi cvor u listu tako da lista bude sortirana po broju stanovnika
void dodaj_sortirano(DRZAVA **pglava, DRZAVA *novi);

// pronalazi i vraca adresu cvora, cija drzava ima najveću gustinu naseljenosti
DRZAVA *najveca_gustina_naseljenosti(DRZAVA *glava);

// pronalazi i vraca adresu cvora, cija država ima najveći BDP po stanovniku
DRZAVA *najveci_bruto_domaci_proizvod_po_stanovniku(DRZAVA *glava);

// dinamicki oslobadja memoriju svakog cvora (koristiti valgrind za proveru) i glavu liste postavlja na NULL
void obrisi_listu(DRZAVA **pglava);

// racuna gustinu naseljenosti tako što kolicnik broja stanovnika i povrsinu zaokruzi na prvu vecu celobrojnu vrednost
int gustina_naseljenosti(int broj_stanovnika, double povrsina);

// izracunava BDP po stanovniku tako sto podeli ukupan BDP sa brojem stanovnika
double bdp_po_stanovniku(double bdp, int broj_stanovnika);
```

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)