

## PRAKTIKUM IZ PROGRAMIRANJA 2

### - domaći zadatak broj 1 -

Sastaviti program na programskom jeziku C koji vrši određenu vrstu obrade nad ulaznim nizom/nizovima. Program treba da:

- učita sve potrebne podatke (dužinu niza, elemente niza, itd.);
- izvrši zahtevanu obradu;
- ispiše sve dobijene rezultate;
- ponavlja prethodna tri koraka dok za dužinu niza/nizova ne bude uneta negativna vrednost;

Zavisno od rednog broja problema, sastaviti **jedan** od sledećih programa, koji:

- u nizu celih brojeva, pronalazi parove brojeva ( $b$ ,  $c$ ), gde je  $b$  najmanji, a  $c$  najveći broj u nizu; brojeve  $b$  i  $c$  treba ispisati na standardnom izlazu, a zatim ih ukloniti iz niza; obrada se ponavlja sve dok niz ima više od jednog elementa;
- vrši proveru i ispisuje da li je prvi zadati niz moguće dobiti rotacijom drugog zadatog niza za odgovarajući broj mesta udesno, kao i broj mesta za koji ga je potrebno rotirati; ukoliko se prvi niz ne može dobiti rotacijom drugog za odgovarajući broj mesta udesno, ukloniti iz prvog niza sve one elemente koji se ne pojavljuju u drugom nizu i ponoviti opisanu proveru;
- vrši aritmetičko pomeranje niza binarnih cifara za zadati broj mesta ulevo ili udesno, a zatim izračunava vrednost binarnog broja formiranog od binarnih cifara u nizu; pre obrade, iz niza ukloniti sve one elemente koji ne predstavljaju binarne cifre 0 ili 1;
- vrši proveru i ispisuje da li niz predstavlja tzv. "testerastu" sekvencu, kao i broj takvih sekvenci u nizu; testerasta sekvenca brojeva je ona sekvenca koja se sastoji od jedne ili više uzastopnih monotono rastućih sekvenci brojeva; nakon provere, iz niza ukloniti testerastu podsekvencu najmanje dužine;
- uklanja sva suvišna, uzastopna pojavljivanja elemenata u zadatom nizu, a zatim vrši proveru da li niz predstavlja bitoničku sekvencu; bitonička sekvenca je sekvenca brojeva koja se sastoji od jedne monotono neopadajuće i jedne monotono nerastuće sekvence brojeva ili obratno ( $x_0 < \dots < x_k > \dots > x_{n-1}$  ili  $x_0 > \dots > x_k < \dots < x_{n-1}$ , gde  $0 \leq k < n$ );
- pronalaži i ispisuje pozicije, a zatim uklanja iz niza celih brojeva sve one elemente čija se binarna reprezentacija čita isto i sleva na desno i sdesna na levo; proveru simetričnosti jednog elementa niza izvršiti na kompletnoj širini celobrojnog tipa koji se koristi;
- računa i ispisuje bit parnosti ([http://en.wikipedia.org/wiki/Parity\\_bit](http://en.wikipedia.org/wiki/Parity_bit)) za svaki element niza celih brojeva, a zatim iz niza ukloni sve one elemente kod kojih je bit parnosti jednak nuli; bit parnosti se računa tako da ukupan broj binarnih jedinica u reprezentaciji broja bude paran;
- vrši proveru i ispisuje da li se prvi zadati niz sadrži u obrnutom poretku u drugom zadatom nizu; ukoliko se prvi niz sadrži u obrnutom poretku u drugom nizu, ukloniti iz drugog niza odgovarajući podniz koji predstavlja prvi niz u obrnutom poretku;

Pri zadavanju niza preko standardnog ulaza, prvo učitati broj elemenata, a zatim i same elemente niza. Kod ispisa niza, ne treba ispisati broj elemenata, već samo elemente niza. Predvideti maksimalni broj elemenata niza (**MAX\_BR\_EL**) pri deklaraciji niza, a u toku rada proveravati da li je uneta dimenzija niza takva da svi elementi koje treba uneti mogu biti smešteni u niz predviđenih dimenzija. Tip elemenata niza odabрати prema potrebama zadatka, odnosno koristiti proizvoljni tip tamo gde to nije suštinski bitno za sam algoritam. Odabрати nekoliko skupova podataka sa kojima će program biti testiran. Svaki primer treba da sadrži ulazne podatke i očekivani izlaz za te podatke.

**Po rešavanju osnovnog problema, izmeniti rešenje zadatka tako da program pre osnovne obrade izvrši sortiranje početnog niza (nizova) po neopadajućem ili nerastućem kriterijumu, ispiše tako sortiran niz (nizove) i nastavi dalje osnovnu obradu. Kriterijum sortiranja korisnik zadaje preko glavnog ulaza.**

**Napomene:**

1. Odbrana prvog domaćeg zadatka je u sredu, 08.04.2015 i četvrtak, 09.04.2015. po rasporedu dostupnom na sajtu predmeta.
2. Formula za redni broj problema **i** koji treba rešavati je sledeća (R – redni broj indeksa, G – poslednje dve cifre godine upisa):  **$i = (R + G) \bmod 8$**
3. Kao rešenje domaćeg zadatka potrebno je na odbrani pokazati sledeće datoteke:
  - **dz1.c**, koja sadrži izvorni tekst osnovnog programa na programskom jeziku C;
  - **dz1\_mod.c**, koja sadrži izvorni tekst izmenjenog programa na programskom jeziku C;

30.03.2015. godine

*sa predmeta*