

PRAKTIKUM IZ PROGRAMIRANJA 1

- domaći zadatak broj 2 -

- Pascal grupa -

Zavisno od rednog broja problema koji se rešava, sastaviti **jedan** od sledećih programa na programskom jeziku Pascal, koji:

0. pronalazi i ispisuje pozicije elemenata niza pozitivnih celih brojeva koji su deljivi brojem 7; za proveru deljivosti koristiti sledeći algoritam: element je deljiv sa 7 ako je broj dobijen oduzimanjem dvostruke vrednosti poslednje cifre elementa od vrednosti ostatka elementa deljiv sa 7; npr. 483: $48 - (3 \times 2) = 42 = 7 \times 6$; algoritam se ponavlja sve dok se ne dobije jednocifren broj; u slučaju da je dobijen broj 0 ili 7, početni broj je deljiv sa 7;
1. proverava da li je ceo broj predstavljen pomoću niza decimalnih cifara deljiv brojem 11; za proveru deljivosti koristiti sledeći algoritam: sabiraju se sve cifre na parnim mestima, kao i sve cifre na neparnim mestima; element je deljiv sa 11 ako je razlika dve izračunate sume deljiva sa 11; npr. 918082: $(9 + 8 + 8) - (1 + 0 + 2) = 22 = 2 \times 11$; algoritam se ponavlja sve dok se ne dobije jednocifren broj; u slučaju da je dobijen broj 0, početni broj je deljiv sa 11;
2. pronalazi i ispisuje pozicije elemenata niza pozitivnih celih brojeva koji su deljivi brojem 8; za proveru deljivosti koristiti sledeći algoritam: broj je deljiv sa 8 ako je zbir vrednosti poslednje cifre i dvostruke vrednosti ostatka broja deljiv sa 8; npr. 156: $(15 \times 2) + 6 = 36$; algoritam se ponavlja sve dok se ne dobije jednocifren broj; u slučaju da je dobijen broj 0 ili 8, početni broj je deljiv sa 8;
3. pronalazi i ispisuje pozicije elemenata niza pozitivnih celih brojeva koji su deljivi brojem 13; za proveru deljivosti koristiti sledeći algoritam: formiraju se blokovi sastavljeni tri susedne cifre, počevši od cifre najmanje težine; sabiraju se svi blokovi na parnim mestima, kao i blokovi na neparnim mestima; broj je deljiv sa 13 ako je razlika dva izračunate sume deljiva sa 13; npr. 2911272: $(2 + 272) - (911) = -637$; nakon izvršavanja algoritma jedanput, proverava se da li je dobijeni broj deljiv sa 13 korišćenjem operacije ostatka pri deljenju;
4. vrši obradu niza decimalnih cifara zadatog broja; iz niza je potrebno ukloniti jednu cifru zadatom broju, tako da rezultat bude najveći broj deljiv brojem 3; broj je deljiv brojem 3 ukoliko mu je zbir cifara deljiv brojem 3, što je dozvoljeno proveriti operacijom ostatka pri deljenju;
5. vrši obradu niza decimalnih cifara zadatog broja; iz niza je potrebno ukloniti jednu cifru zadatom broju, tako da rezultat bude najveći broj deljiv brojem 2^n , gde se n unosi sa standardnog ulaza i važi $n \geq 1$; broj je deljiv sa 2^n ukoliko je broj koji čine poslednjih n cifara deljiv brojem 2^n , što je dozvoljeno proveriti operacijom ostatka pri deljenju;
6. vrši obradu niza decimalnih cifara zadatog broja; niz je potrebno rotirati udesno odgovarajući broj puta dok se ne dobije najveći mogući broj koji je deljiv brojem 8; rotacija udesno se dobija premeštanjem najniže cifre na mesto najviše i tako redom; broj je deljiv brojem 8 ukoliko je broj koji čine poslednje tri cifre deljiv brojem 8, što je dozvoljeno proveriti operacijom ostatka pri deljenju;
7. na osnovu niza celih brojeva formira novu listu koja će za svaki element polaznog niza sadržati vrednost njegove aditivne perzistencije; aditivna perzistencija broja je jednaka broju puta koji treba ponoviti operaciju sabiranja cifara na dobijeni rezultat sabiranja da bi se dobio jednocifren broj; npr. aditivna perzistencija broja 2718 je jednaka 2: $(2 + 7 + 1 + 8) = 18$, $(1 + 8) = 9$;

Programi treba da učitaju sve potrebne podatke, zatim izvrše zahtevanu obradu i, na kraju, ispišu rezultate te obrade. Pri zadavanju niza, prvo učitati broj elemenata, a zatim i same elemente niza. Pri tome, proveriti da li je veličina niza dovoljna za smeštanje svih elemenata. Kod ispisivanja niza, ne treba ispisivati broj elemenata, već samo elemente niza. Predvideti maksimalni broj elemenata niza (**MAX_BR_EL**) pri deklaraciji niza. Program treba da ponavlja ceo postupak sve dok korisnik ne unese neku neodgovarajuću vrednost za neki od ulaznih podataka (npr.

nepozitivan ili previše veliki broj za broj elemenata niza). Tip elemenata niza odaberi prema potrebama zadatka, odnosno koristiti proizvoljni tip tamo gde to nije suštinski bitno za sam algoritam.

Kandidati koji pre odbrane na posebnom listu papira ili datoteci nemaju spremna makar tri suštinski različita test primera neće moći da pristupe odbrani. Svaki primer treba da sadrži ulazne podatke i očekivani izlaz za te podatke.

Dodatan zahtev: po uspešnom rešavanju osnovnog zadatka, napraviti novi program koji predstavlja modifikaciju osnovnog programa. Nakon završetka osnovne obrade iz početnog niza treba ukloniti sve lokalne ekstremne vrednosti. Nakon uklanjanja elemenata ispisati niz i njegovu novu dužinu.

Napomene:

1. Odbrana drugog domaćeg zadatka je u sredu, 27.11.2019. Tačni termini za sve studente su objavljeni 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:
 - dz2.PAS, koja sadrži izvorni tekst osnovnog programa na programskom jeziku Pascal;
 - dz2_mod.PAS, koja sadrži tekst modifikovanog programa;
 - dz2_test.txt, koja sadrži primere očekivanih ulaza i izlaza programa;

20.11.2019. godine

sa predmeta