

PRAKTIKUM IZ PROGRAMIRANJA 1

- domaći zadatak broj 2 -

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

0. određuje uniju dva skupa; skup se predstavlja pomoću rastuće uređenog niza celih brojeva; obezbediti prilikom učitavanja da se u skup ne može više puta uneti isti element;
1. vrši pretvaranje niza uglova zadatih stepenima u niz uglova zadatih u radianima, a zatim ispisuje vrednosti i pozicije svih oštih uglova u nizu; elementi niza su realni brojevi;
2. u nizu celih brojeva $a[i]$ pronalazi i ispisuje sve uređene trojke (p, q, r) , takve da važi $p = q + r$; q i r su različiti elementi od p ;
3. u nizu pozitivnih realnih brojeva $a[i]$ pronalazi i ispisuje sve parove (p, q) čija je aritmetička sredina veća ili manja od podjednagog broja preostalih elemenata u nizu;
4. pronalazi i ispisuje one elemente niza celih brojeva koji su jednaki i -tom Fibonačijevom broju, gde je i indeks elementa u nizu (uslov $a[i] = F_i$); Fibonačijevi brojevi se računaju po formuli $f_0 = 0$, $f_1 = 1$, $f_n = f_{n-1} + f_{n-2}$;
5. računa zbir cifara za svaki element u nizu celih brojeva, a zatim ispisuje vrednosti i pozicije svih onih elemenata niza čiji zbir cifara nije prost broj;
6. za svaki element zadatog niza $a[i]$ celih brojeva utvrđuje broj pojavljivanja u nizu celih brojeva $b[i]$, a zatim ispisuje sve pozicije u nizu $b[i]$ elementa sa najvećim brojem pojavljivanja;
7. u nizu brojeva $a[i]$ pronalazi i ispisuje vrednosti pozicije svih elemenata koji su veći od aritmetičke sredine b prethodnih i b narednih elemenata u nizu; ukoliko ne postoji dovoljno prethodnih ili narednih elemenata uzeti onoliko elemenata koliko postoji; b je pozitivan ceo broj koji se unosi sa standardnog ulaza;

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 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.

Važno: po uspešnom rešavanju osnovnog zadatka, napraviti novi program koji predstavlja modifikaciju osnovnog programa. Nakon završetka osnovne obrade iz početnog niza/nizova treba ukloniti sva pojavljivanja minimalnog i maksimalnog elementa u nizu (zadaci 1, 3, 5, 7), odnosno svih neparnih elemenata u nizu (zadaci 0, 2, 4, 6). Nakon uklanjanja elemenata ispisati niz i njegovu novu dužinu.

Napomene:

1. Odbrana drugog domaćeg zadatka je u četvrtak, 30.11.2017. 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;