

U direktorijumu **Rad** koji se nalazi na desktopu kreirati direktorijum **Ime\_prezime\_indeks** i u okviru njega kreirati **c** fajlove koji sadrže programe koji rešavaju sledeće probleme, pri čemu ime fajla MORA DA ODGOVARA „imenu“ zadatka.

### **Prvi:**

Majka želi da organizuje svoje sinove Pericu i Janka kako bi joj kupili potrebne lekove. Ona pravi isti spisak lekova za njih (lekovi se označavaju preko jedinstvenog ID-a koji je ceo broj), ali pravi dva spiska apoteka. Apoteke su uređene u odnosu na njihovu kuću od najbliže ka najudaljenijoj. Svaka apoteka sadrži naziv (više reči), broj lekova, niz lekova koje poseduje (niz celih brojeva) i lokaciju predstavljenu preko dve koordinate X i Y (realni brojevi). Smatra se da je pozicija kuće (0,0).

- Na ulazu se najpre zadaje broj N, a zatim podaci o apotekama za Pericu. Zatim se unosi M i podaci o apotekama za Janka. Voditi računa o tome da se apoteke uređuju prema udaljenosti od njihove kuće.
- Ispisati formirane liste apoteka.
- Zatim se unosi K i niz ID-jeva lekova koji se nalaze na spisku.
- Perica i Janko kreću u kupovinu lekova prema svojim listama apoteka. Oni su zadatak shvatili kao malu igru ko će biti efikasniji i kupiti više lekova ali su uveli neka dodatna pravila. Mobilnim porukama obavestavaju jedan drugog o tome koju apoteku običu i ukoliko imaju neku istu apoteku na spisaku dozvoljeno je da u nju uđe i kupi lekove samo onaj koji je do nje prvi stigao. U jednoj apoteci kupuju sve lekove koji se nalaze na spisku a ta apoteka ih poseduje. Lekovi kupljeni u jednoj apoteci ne traže se dalje. Štampati ko je prvi kupio sve lekove. Ukoliko su obišli sve apoteke sa svoje liste a nisu kupili sve lekove, štampati ko je kupio više lekova i ID-jeve lekova koje je kupio.
- Narednog dana žele da ponove igru, ali samo ažuriraju svoje liste, ne kupuju lekove. Liste sada nemaju preklapanja, potpuno su različite. Ako je bilo istih apoteka, ona ostaje u onoj listi u kojoj bi se pre do nje stiglo, iz druge se briše.
- Ispisati ažurirane liste apoteka.

NAPOMENA. Pri unosu poštovati navedeni redosled podataka.

### **Drugi1:**

Najveća panoramska vrteška na svetu „London eye“ ima različite ponude za svoje turiste. Osnovna podela je na turističku (koja podrazumeva jednu vožnju) i privatnu (koja podrazumeva zakup na određeno vreme). Sama vrteška se sastoji od 20 kapsula u koju najviše može da stane 10 osoba. Turisti već znaju da ukoliko organizuju grupne posete njihova grupa može imati najviše 10 osoba. Vrteška se privremeno zaustavlja svaki put kada kapsula dodje u položaj da može da prima turiste, u tom položaju ostaje 2min, a sledeća kapsula dolazi u taj položaj za 5min. Ukoliko je kapsula privatno rezervisana na određeno vreme, i u trenutku zaustavljanja to vreme nije isteklo iz nje ne izlaze turisti i ne ulaze novi.

Potrebno je napisati program koji rešava problem raspoređivanja turista u okviru kapsula. Za svaku kapsulu se pamti koliko se turista trenutno nalazi u njoj (ceo broj), podatak u koju svrhu je trenutno zakupljena (slovo T-turistički ili slovo P-privatno). Ukoliko je zakupljena privatno postoji još jedan podatak a to je na koliko minuta (ceo broj).

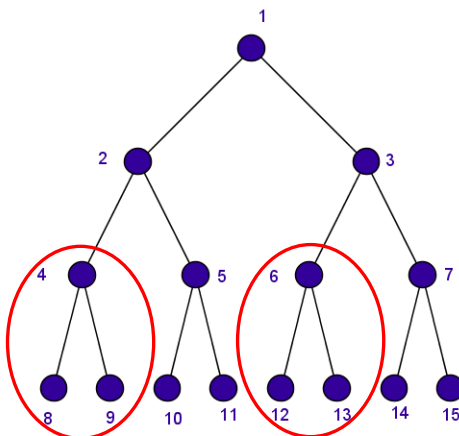
- Na ulazu se redom unose turisti (sve dok postoje). Svaki turista ima kartu koja ima svoj kod koji predstavlja niz od 10 karaktera. Poslednji karakter predstavlja oznaku T ili P (tip posete), prethodni karakter predstavlja oznaku G ili S, u zavisnosti da li je u pitanju grupna ili samostalna poseta. Ukoliko je u pitanju G-grupna poseta prva tri karaktera koda predstavljaju ID grupe, kod S-samostalne posete ti karakteri koda nemaju značenje. Ukoliko je poseta P-privatna 5, 6 i 7 karakter zajedno predstavljaju na koliko minuta je kapsula rezervisana. Turisti redom ulaze u kapsule pri čemu se vodi računa o tome da grupne posete idu u istu kapsulu, i ukoliko grupa turista ne može da stane u delimično popunjenu kapsulu, ona ostaje takva a oni ulaze u sledeću. Grupa ne podrazumeva da se samo ona nalazi u kapsuli, ukoliko može još neko da stane on ulazi u tu kapsulu. Imati u vidu da poseta može biti istovremeno grupna, privatna zakupljena na određeno vreme.
- Ispisati formiranu kružnu listu sa podacima o kapsulama nakon prvog kruga.
- Nastaviti postupak raspoređivanja turista sve dok ih ima. Ukoliko je neka kapsula bila privatna, u nekom od narednih krugova to ne mora biti. Štampati nakon koliko vremena od početka raspoređivanja turista je u kapsulu ušao poslednji turista.
- Štampati kružnu listu u poslednjem krugu, posle ulaska poslednjeg turista.

NAPOMENA. Pri unosu poštovati navedeni redosled podataka.

### Drugi2:

Na ulazu se zadaje ceo broj N, a zatim N reči. Formirati binarno stablo čiji su elementi strukture koje sadrže REC (uneta reč) i ZBIR\_ASCII (ceo broj koji predstavlja zbir ASCII vrednosti karaktera na neparnim pozicijama, smatra se da je prvi karakter reči na neparnoj poziciji) i koje je uređeno u odnosu na ZBIR\_ASCII. Ispisati podatke formiranog binarnog stabla ali tako da se prvo ispišu oni kod kojih je zbir ASCII vrednosti karaktera na neparnim pozicijama manja.

- Napisati rekursivnu funkciju koja za uneto potpuno stablo štampa svaku drugu trojku čvora sa njegovim naslednicima koji su listovi.



NAPOMENA. Pri unosu poštovati navedeni redosled podataka.

### Treći:

Korišćenjem dinamičkih struktura podataka Crveno-Crna(max 20 poena) stabla (Ukoliko se zadatak reši korišćenjem AVL stabla maksimum je 17 poena.) rešiti sledeći problem:

Program treba da implementira podsetnik za rođendane. Program sa ulaza čita niz linija oblika:

Ime Prezime DD.MM.YYYY.

Odnosno, za svaku osobu se unosi datum rođenja. Smatra se da su svi učitani podaci ispravni i zaista predstavljaju datume.

Za svaku osobu se pamti ime(jedna reč), prezime(jedna reč), dan(ceo broj), mesec(ceo broj) i godina rodjenja(ceo broj), Broj\_istih(ceo broj, koji predstavlja koliko osoba rođendan slavi tog datuma). Stablo se uređuje prema datumu rođenja.

- Nakon završenog rasporeda osoba štampati podatke o njima od najmlađih ka najstarijim.
- Ažurirati kreirano stablo izbacivanjem svih osoba mlađih od 10 godina u odnosu na današnji datum (Brisanje bez kreiranja novog stabla donosi bonus od 3 poena).
- Štampati podatke o osobama iz ažuriranog stabla od najmlađih ka najstarijim.
- Napisati rekursivnu funkciju koja **za dati koren stabla** i datum pronalazi osobu čiji je rođendan najbliži datom datumu. Funkcija vraća NULL, ako je stablo prazno, odnosno adresu osobe čiji je rođendan najbliži datom datumu.
- U glavnom delu programa štampati podatke o osobi čiji je rođendan najbliži datom datumu.