

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:

- Potrebno je pomoći Nindža kornjačama da rekonstruišu kanalizacionu mrežu Njujorka.
- U samom startu Nindža kornjače dostavljaju trenutno stanje mreže koja se sastoji od N cevi koje se nadovezuju jedna na drugu. U nekim slučajevima su čak i loše nadovezane jer je nemoguće jednu staviti u drugu, pa su se razdvojile i dolazi do nemogućnosti da se jedna cev smesti u drugu (to se određuje na osnovu dimenzija). Za svaku cev se pamte sledeći podaci:
 - ✓ ID cevi (ceo broj)
 - ✓ Indikator oštećenja (ceo broj: 1 – oštećena cev, 0 – neoštećena)
 - ✓ Vrsta cevi (više reči)
 - ✓ Prečnik cevi (realan broj – u prečnik spada i debljina cevi)
 - ✓ Debljina cevi (realan broj)
 - ✓ Dužina cevi (realan broj)
- Prvo se dostavlja broj cevi N a zatim redom podaci o cevima. Podaci se čuvaju u okviru jednostruko povezane liste u redosledu kako se unose.
- Ispisati kreirano stanje kanalizacione mreže.
- Eksperti su napravili plan rekonstrukcije kanalizacione mreže i dostaviće ga Nindža kornjačama. Potrebno je uraditi sledeće:
 - Ukoliko dve susedne cevi zbog svojih dimenzija ne mogu da se smeste jedna u drugu, potrebno je cev manjeg prečnika zameniti novom “PLASTICNOM” novih dimenzija. Problem određivanja dimenzija rešite tako da se nova cev može smestiti u obe svoje susedne cevi, ali da pri tom labav prostor ne postoji sa jedne strane.
 - Ispisati trenutno stanje plana kanalizacione mreže.
 - Ukoliko je jedna cev oštećena, a njene susedne cevi nisu, ona se mora zameniti novom “PLASTICNOM”, istih dimenzija.
 - Ispisati trenutno stanje plana kanalizacione mreže.
 - Ukoliko je više uzastopnih cevi oštećeno one se menjaju jednom “PLASTICNOM” cevi dužine jednake zbiru dužina svih cevi koje se menjaju. Vodi se računa da se cev mora nadovezati na obe strane. Problem određivanja dimenzija rešite tako da se nova cev može smestiti u obe svoje susedne cevi, ali da pri tom labav prostor ne postoji sa jedne strane.
- Ispisati kreirano stanje kanalizacione mreže nakon rekonstrukcije.

NAPOMENA: Nije dozvoljeno proširivati strukturu, niti koristiti nizove.

Drugi1:

Organizuje se vojna igra koja se sastoji od kružnog poligona koji ima **N** mesta. Svako mesto ima ID (ceo broj koji se ne učiva, već se redom dodeljuje 1, 2, 3... za svako učtano mesto), indikator za samouništenje (ceo broj: 1 znači da grad eksplodira, 0 znači da se u gradu se može boraviti), indikator da li je igrač u gradu (ceo broj: 1 jeste, 0 nije – u startu su gradovi prazni) i ako jeste ID igrača (ceo broj – opet u startu nema igrača). Ukoliko se u gradu može boraviti, onda se iz njega može i kretati do drugih gradova, tako da se definišu još dva podatka: smer kretanja (ceo broj: -1 za levo, 1 za desno) i broj gradova koji se preskače (ceo broj). U svrhu što bolje organizacije koristi se **kružna dvostruko povezana lista**. U programu se najpre unosi broj N a zatim podaci o mestima, na osnovu kojih se formira kružna dvostruko povezana lista. Štampati podatke formirane liste.

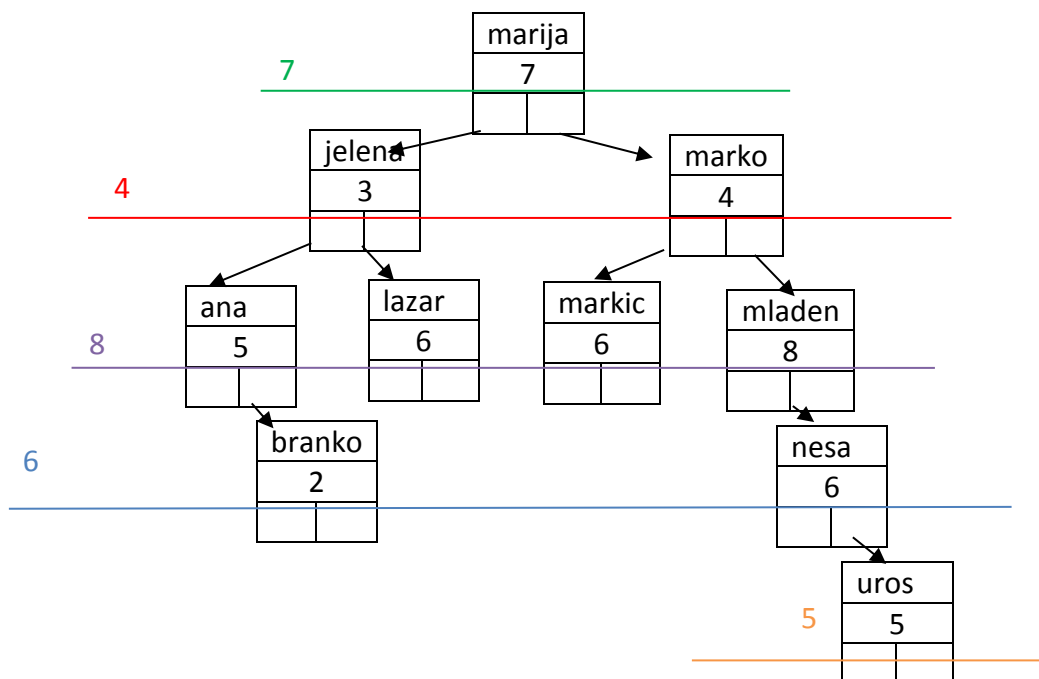
Zatim se vrši postavljanje M igrača u određena mesta, tako što se navode redom po dva broja koji predstavljaju ID igrača i ID mesta u kome se smešta igrač. Ukoliko se u nekom mestu koje se samouništava odmah smesti neki igrač, automatski nestaju i mesto i taj igrač. Nakon postavljanja igrača u gradove štampati podatke trenutne liste.

Igra traje 3 koraka (svaki igrač može da se kreće najviše 3 puta iz mesta u mesto). Igru započinje od prvog zauzetog mesta i igrači se kreću na osnovu smera i broja koraka koji su definisani za mesto u kome stoji. Ukoliko igrač treba da dođe u mesto gde se već nalazi neki igrač, to mesto se uništava zajedno sa igračima koji su u njemu, ukoliko dođe u slobodno mesto koje se samouništava opet nestaju i igrač i to mesto. Nakon svakog koraka štampati stanje liste.

NAPOMENA. Pri unosu poštovati navedeni redosled podataka.

Drugi2:

Na ulazu se zadaje ceo broj N a zatim N **Prijava**. Elementi strukture **Prijava** su NAZIV_IGRACA(jedna reč) i PRIJAVLJENI_BROJ. Formirati binarno stablo čiji su elementi **Prijava** i koje je uređeno u odnosu na NAZIV_IGRACA. Ispisati kako izgleda formirano stablo ali tako da se podaci o prijavama ispišu u alfabetskom poretku (A-Z). Napisati funkciju kojom se određuju maksimalna prijavljena vrednost na svakom nivou stabla.



Treći:

Korišćenjem dinamičkih struktura podataka stabla (Crveno-Crna maksimum poena je 22, ukoliko se zadatak reši korišćenjem AVL stabla maksimum je 19 poena) sastaviti program koji treba da organizuje licitiranje proizvoda. Za svaki **proizvod** se pamti šifra proizvoda (ceo broj), broj licitiranih cena (ceo broj u startu 0) i niz licitiranih cena koja je u startu prazan (niz celih brojeva). Najpre se unosi broj proizvoda N a zatim šifre proizvoda i formira se binarno pretrazivačko stablo uređeno prema šifri proizvoda.

- Nakon završenog rasporeda proizvoda štampati podatke o njima od onog sa najmanjom šifrom do onog sa najvećom šifrom. Štampati sve podatke o proizvodima. Zatim korišćenjem date funkcije za štampu po nivoima, odštampati kreirano stablo, prikazujući samo šifru proizvoda.
- Zatim se održava licitacija koja traje sve dok se ne unese 0 za šifru proizvoda. Svaki potencijalni kupac kaže šifru proizvoda i ponuđenu cenu i sve se registruje u odgovarajućem nizu licitiranih cena nekog proizvoda.
- Nakon završenog rasporeda licitiranih cena proizvoda štampati sve podatke o njima od onog sa najmanjom šifrom do onog sa najvećom šifrom.
- Zatim se vrši izbacivanje svih nelicitiranih proizvoda. (Brisanje bez kreiranja novog stabla donosi bonus od 3 poena.)
- Nakon završenog izbacivanja nelicitiranih proizvoda štampati podatke o njima od onog sa najmanjom šifrom do onog sa najvećom šifrom. Štampati sve podatke o proizvodima. Zatim korišćenjem date funkcije za štampu po nivoima, odštampati kreirano stablo, prikazujući samo šifru proizvoda.
- Napisati rekursivnu funkciju kojom se određuje koja cena se najviše puta pojavljivala kao licitirana cena proizvoda.