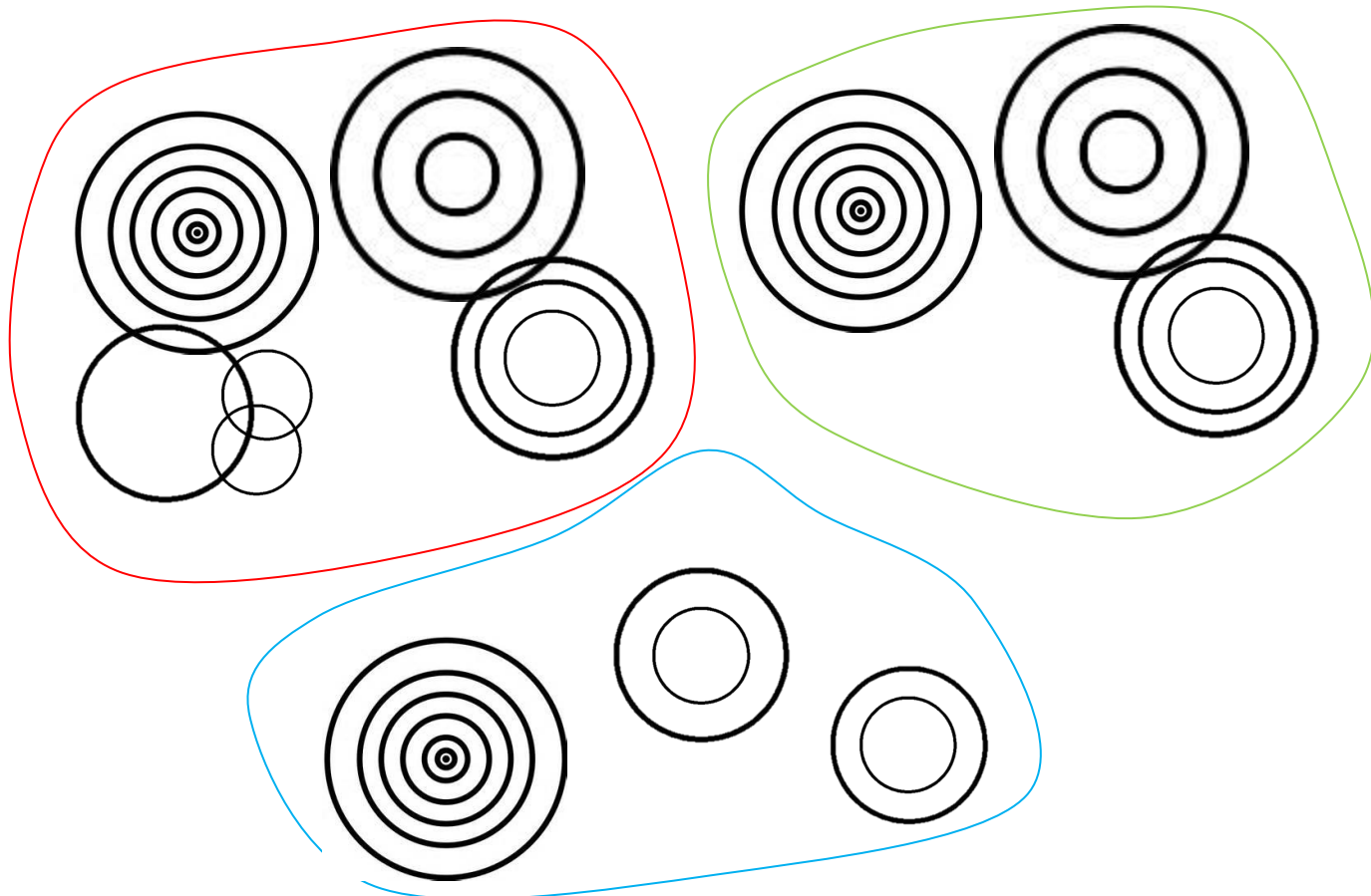


U direktorijumu **Rad** koji se nalazi na desktopu kreirati direktorijum **Ime_prezime_indeks** i u okviru njega kreirati **c** fajlove koji sadrže programe koji rešavaju sledeće probleme, pri čemu ime fajla MORA DA ODGOVARA „imenu“ zadatka.

Prvi:

- Potrebno je demonstrirati igru „**Kruzimo**“ u kojoj učestvuje **n** kružnica. Za svaku KRUŽNICU se pamte sledeći podaci:
 - ✓ X koordinata (realan broj)
 - ✓ Y koordinata (realan broj)
 - ✓ Poluprečnik (realan broj)
 - ✓ ID (ceo broj)
 - ✓ Broj zajedničkih (ceo broj)
 - ✓ niz IDa (celi brojevi - neće biti više od 20)
- Igra se organizuje tako što se prvo unosi broj kružnica N, a zatim podaci o njima na osnovu kojih se kreira igra. Kružnice kako se unose zauzimaju poslednju poziciju u listi. Za svaku kružnicu se najpre unose koordinate centra (X i Y), poluprečnik kružnice, ID se dodeljuje kao redni broj učitane kružnice (prva kružnica ima ID 1). Broj zajedničkih se inimizuje na nulu i predstavlja broj celih brojeva u nizu ID-jeva koji je u startu prazan.
- Ispisati kreiranu igru sa **svim** podacima o kružnicama. Napraviti univerzalnu funkciju za ispis koja će moći pri kasnijim pozivima da ispisuje i podatak broj zajedničkih kao i niz ID-jeva.
- Ažurirati podatke o kružnicama tako da svaka kružnica sadrži u nizu ID-jeva sve ID kružnica sa kojima se seče. Kružnice se seku ako imaju dve zajedničke tačke. Broj zajedničkih sadrži broj kružnica sa kojima se data kružnica seče. Ispisati kreiranu igru sa **svim** podacima o kružnicama.
- Štampati sve grupe koncentričnih kružnica odvojene linijom ----- . Jedna kružnica ne čini grupu. Možete napraviti listu grupa, koristiti istu strukturu, samo što niz ID-jeva sadrži ID kružnica koje pripadaju grupi.
- Štampati sve slobodne grupe koncentričnih kružnica odvojene linijom ----- . Grupa se smatra slobodnom ukoliko nijedna njena kružnica nema zajedničkih tačaka sa drugim kružnicama. Kružnica iako treba da pripada nekoj grupi, sada joj ne pripada ukoliko seče kružnicu neke druge grupe (dovoljno je izbrisati jednu od njih, ali nije greška ukoliko se uklone obe kružnice). Jedna kružnica ne čini grupu.



Drugi1:

U dvostruko povezanu listu LIST smeštaju se podaci o procesima koji čekaju na izvršavanje: identifikator procesa (ceo broj), prioritet procesa (ceo broj), vlasnik procesa (jedna reč) i vreme pokretanja procesa (jedna reč). Lista je uređena prema prioritetu procesa (proces sa najvećim prioritetom nalaze se na početku, ukoliko imaju isti prioritet onaj sa manjim identifikatorom procesa nalazi se pre). U programu se najpre unosi broj n a zatim podaci o n procesa. Štampati podatke o procesima od onog sa najvećim prioritetom ka manjima. Potrebno je napisati funkciju koja će datom procesu s identifikatorom **idProcesa** promeniti prioritet u **prioritet** i ponovno urediti listu prema prioritetu procesa. Funkcija treba imati prototip:

```
void SetAndSort(LIST** lista, int idprocesa, int prioritet);
```

U glavnom delu procesa uneti **idProcesa** i **prioritet** koji mu se postavlja datom funkcijom. Štampati podatke o procesima od onog sa najvećim prioritetom ka manjima nakon promene prioriteta datom procesu.

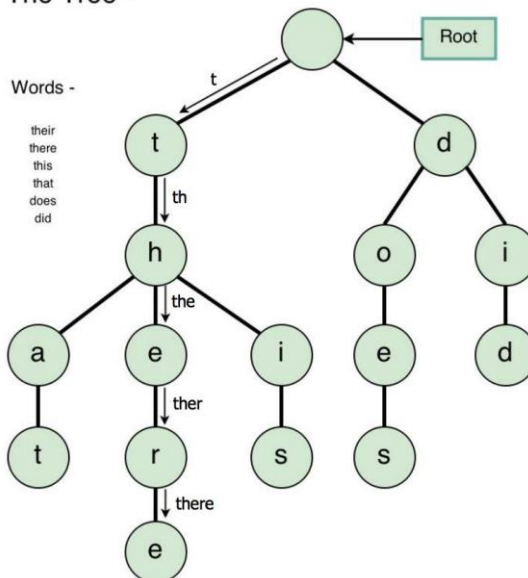
NAPOMENA. Pri unosu poštovati navedeni redosled podataka.

Drugi2:

Na ulazu se zadaje ceo broj N, a zatim N reči. Formirati **trie ili digitalno stablo** čiji su elementi strukture koje mogu da sadrže jedno slovo engleskog alfabeta (nije neophodno) i niz pokazivača ka mogućim ostalim čvorovima. Smatrati da ima 26 slova. Ovo stablo je pogodno za čuvanje teksta. Primer je prikazan na slici.

Napisati funkciju koja za datu reč sa ulaza proverava da li se nalazi u stablu.

Trie Tree -



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:

Policija je uhvatila opasnog hakera koji je pokušao (ali nije uspeo ☹) da probije zaštitinu šifru IMI-ja. U njegovom računaru je pronađeno mnogo šifrovanih poruka razvojenih blanko karakterima. Stručnjaci za šifrovanje su otkrili da jedna maskirana šifra u sebi sadrži niz šifara za logovanje na neke računare. Šifre koje su skrivene u poruci se sastoje samo od cifara 1,2,3,4 i 5 i napisane su obrnuto. Recimo broj 123 se šifruje kao 321 . Cifra 6 označava kraj jedne šifre dok cifre 7,8,9 predstavljaju šum i treba ih ignorisati.

Primer : K = 326784456 Izlaz : 544 23

K = 167288368926 Izlaz: 2 32 1

Ako je jedna pročitana šifra K, na datom primeru se može videti koje ona šifre sve sadrži.

Za svaku SIFRU se pamti njena vrednost i broj ponavljanja. Stablo se uređuje na osnovu vrednosti šifre.

- Nakon završenog rasporeda šifara štampati podatke o njima od najveće na najmanjoj.
- Napisati rekurzivnu funkcija koja **za dati koren stabla i pravu šifru** (ceo broj) pronalazi šifru čija je vrednost najbliža pravoj šifri. Funkcija vraća NULL, ako je stablo prazno, odnosno adresu šifre čija je vrednost najbliži datoj pravoj šifri.
- Ažurirati kreirano stablo izbacivanjem šifre koja je najbliža unetom celom broju K.
- Ispisati podatke o šiframa od najveće na najmanjoj nakon ažuriranja stabla.
- U glavnom delu programa štampati podatke o šifri čija je vrednost najbliža unetom celom broju K (u primeru prvo se unosi broj K=45 a zatim se čitaju šifre).

45 326784456 167288368926 48974561626 123456 8886 16 1288886 811648746 755576 5758596	Pamte se sledeće podaci (samo su navedene prema unosu ne prema rasporedu u stablu): 544 2 23 1 2 2 32 1 1 3 54321 1 21 1 11 1 44 1 555 2
---	---