

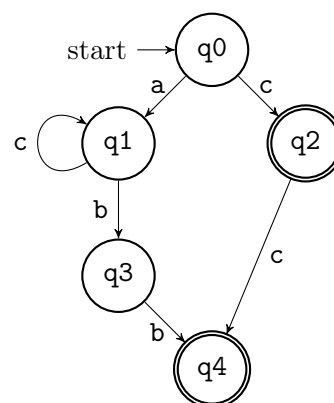

```

Sprejemnik bojan = new Sprejemnik(krvavec); // tudi bojan posluša krvavec
Sprejemnik cvetka = new Sprejemnik(nanos); // cvetka ima raje nanos
nanos.oddaj('c'); // to sprejme cvetka
krvavec.oddaj('d'); // to sprejmeta ana in bojan
System.out.println(ana.odZadnjic()); // bd
System.out.println(cvetka.odZadnjic()); // c
krvavec.oddaj('e'); // to sprejmeta ana in bojan
System.out.println(ana.odZadnjic()); // e
System.out.println(bojan.odZadnjic()); // de
System.out.println(bojan.odZadnjic()); // (prazno)
System.out.println(ana.koliko()); // 3
System.out.println(bojan.koliko()); // 2

```

Lahko predpostavite, da vsakemu oddajniku pripada kvečjemu po 1000 sprejemnikov in da vsak sprejemnik v času svojega obstoja prejme kvečjemu po 1000 znakov. V 60% skritih testnih primerov je vsak oddajnik povezan s kvečjemu enim sprejemnikom, v polovici od teh primerov pa ni nobenega klica metode `odZadnjic`.

- ④ *Končni avtomat* je stroj, ki po vrsti bere znake vhodnega niza (*besede*) in pri tem spreminja svoje stanje. Na primer, avtomat, ki ga ponazarja diagram na desni (`Test01.java`, `Test04.java`), je na začetku v stanju `q0` (*začetno stanje*), po branju besede `acb` pa pristane v stanju `q3` ($q0 \xrightarrow{a} q1 \xrightarrow{c} q1 \xrightarrow{b} q3$). Stanji `q2` in `q4` sta *sprejemni stanji*; če avtomat po branju besede pristane v enem od njih, pravimo, da besedo *sprejme*. Naš avtomat potemtakem sprejme besede `c`, `cc`, `abb`, `acbb`, `accbb`, `accbb` itd.



Končni avtomat je predstavljen kot objekt razreda `Avtomat` v datoteki `Cetrta.java`, ki vsebuje sledeče atribute:

```

private String zacetnoStanje;
private Set<String> sprejemnaStanja;
private Map<String, Map<Character, String>> prehodi;

```

Slovar `prehodi` pove, v katero stanje preide avtomat pri podanem izhodiščnem stanju in podanem znaku. V našem primeru bi klic `prehodi.get("q1").get('b')` vrnil niz `q3`, klic `prehodi.get("q2").get('a')` pa vrednost `null`, saj ta prehod ne obstaja. Klic `prehodi.get("q4")` bi vrnil prazen slovar.

Razred `Avtomat` dopolnite s sledečimi metodami:

- [32%] `public boolean jeSprejemno(String stanje)`
Vrne `true` natanko v primeru, če je podano stanje sprejemno.
- [34%] `public String kam(String beseda)`
Vrne stanje, v katerem se avtomat nahaja po branju podane besede. Če kateri od prehodov ne obstaja, naj metoda vrne `null`.
- [34%] `public static Avtomat zaBesedo(String beseda)`
Vrne končni avtomat, ki sprejme samo podano besedo. Namig: i -to stanje naj bo kar `Integer.toString(i)`.
Javni testni razredi preverjajo to metodo z vašima, skriti pa z našima (zanesljivo pravi) implementacijama metod `jeSprejemno` in `kam`.