

1. naloga (20 točk, čas reševanja: 20 minut)

V razredu `Naloga31.java` napiši metodo

```
String poiisciUjemanje(String prvi, String drugi),
```

ki poišče in vrne najdaljše zaporedje znakov, ki se pojavi tako v prvem kot v drugem nizu. Metoda `poiisciUjemanje()` naj ne bo statična!

V razredu `Naloga31.java` napiši tudi metodo `main()`, ki pokliče metodo `poiisciUjemanje()` s prvima dvema argumentoma ukazne vrstice in rezultat izpiše na zaslone.

Namig: nalogo lahko rešiš tako, da v drugem nizu iščeš vse možne podnize prvega niza.

Primer. Ob klicu programa

```
java Naloga31 "Danes je lep dan" "Dandanes je lepo"
```

naj se na zaslon izpiše:

Najdaljsi skupni podniz: 'anes je lep'.

2. naloga (25 točk, čas reševanja: 30 minut)

Preprost logotip lahko narišemo kot sliko velikosti 5x5. Primer tovrstnega logotipa je prikazan na spodnji levi sliki.

Podatke o logotipi (katere slikovne točke so vidne in katere ne) lahko na učinkovit način shranimo v 25 zaporednih bitih in sicer tako, da prvih 5 bitov predstavlja 1. vrstico logotipa, drugih 5 bitov drugo vrstico logotipa, in tako naprej. Celoten logotip lahko torej shranimo v eni spremenljivki tipa `int`.

Primer: logotip na levi sliki predstavlja število 32837969, oziroma

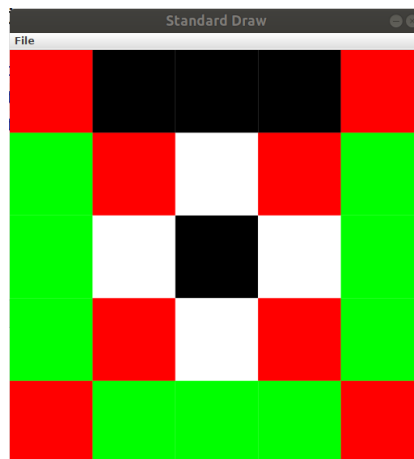
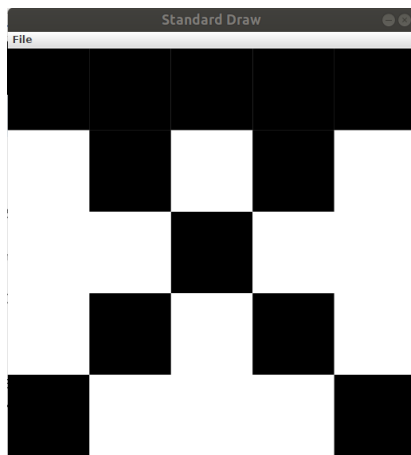
```
int logotip = 0b1111101010001000101010001.
```

Napiši program, ki prejme kot argumenta dve števili tipa `int` in s pomočjo razreda `StdDraw` nariše ustrezna logotipa. Nariše ju tako, da kvadratke, ki pripadajo samo prvemu logotipu, pobarva rdeče, kvadratke, ki pripadajo samo drugemu logotipu, pobarva zeleno, skupne kvadratke pa pobarva črno.

Primer: klic

```
java Naloga32 32837969 15259182
```

naj izriše desno sliko.



Namig: če z ukazom `StdDraw.setScale(0,5)` nastaviš skalo risalnega območja na $[0,4] \times [0,4]$, potem en kvadratega logotipa na lokaciji (i,j) narišeš z ukazom

```
StdDraw.filledRectangle(i+0.5,j+0.5,0.5,0.5);
```

3. naloga (25 točk, čas reševanja: 30 minut)

Datotečni format WAV (*Waveform File Format*) je standardni format za zapis zvočnih datotek. Datoteko wav sestavlja 44 bajtov dolga glava, kateri sledijo podatki zvočnega zapisa. Glava vsebuje naslednje podatke:

Pozicija (bajti)	Primer vrednosti	Hex vrednost	Opis
1 - 4	RIFF	52 49 46 46	Označuje kontejner RIFF. Vsak znak (oz. ASCII koda znaka) zavzame 1 bajt (0x52 je znak 'R').
5 - 8	2.469.640	08 af 25 00	Celo število (32-bitni integer), ki določa število bajtov, ki sledijo (torej je enako velikosti datoteke, zmanjšane za 8).
9 - 22			parametri zapisa
23-24	2	02 00	Celo število (16-bitni integer), ki pove število kanalov .
25-28	44.100	44 ac 00 00	Celo število (32-bitni integer), ki pove frekvenco vzorčenja v Hz (tj. število vzorcev na sekundo, npr. 44100 za CD, 48000 za DAT).
29-44			parametri zapisa

Vsa števila v wav datoteki so zapisana kot *little endian*, kar pomeni, da je najprej zapisan najmanj pomemben bajt. Tako lahko, na primer, "število bajtov, ki sledijo" (zapisano na poziciji 5-8) ugotovimo takole: 08 af 25 00 --> 0025af08₁₆ --> 2.469.640₁₀.

- Napiši metodo `preberiPodatkeWav()`, ki kot parameter prejme ime zvočne datoteke in vrne podatke o številu kanalov in frekvenci vzorčenja.
- Napiši razred `IzjemaWavFormata` ter metodo `preberiPodatkeWav()` spremeni tako, da bo sprožila izjemo, če datoteka ni v pravilnem formatu wav. Pri tem preveri oznako `RIFF` ter ujemanje zapisanega števila bajtov (na poziciji 5-8) z dejansko velikostjo datoteke. Če kje pride do odstopanja, naj metoda sproži izjemo s sporočilom, kje je napaka (v zapisu `RIFF` ali v dolžini datoteke).
- Napiši program `Naloga33.java` z metodo `main()`, v kateri pokličeš metodo za branje podatkov zvočne datoteke `preberiPodatkeWav()` ter izpišeš podatke o frekvenci vzorčenja in številu kanalov. Pri tem ustrezno obravnavaj morebitne izjeme.

4. naloga (30 točk, čas reševanja: 40 minut)

Geografske podatke hranimo v dveh datotekah. V prvi datoteki so zapisani podatki o državi (kratica države, ime države, število prebivalcev države), v drugi pa podatki o glavnih mestih (kratica države, ki ji mesto pripada, glavno mesto, število prebivalcev mesta). Vsak zapis je v svoji vrstici, posamezni podatki v vrstici so ločeni s podpičjem.

Preberi podatke obeh datotek (imeni datotek sta podani kot prvi in drugi argument ob klicu programa) in za vsako državo izpiši ime države, glavno mesto in delež prebivalcev v glavnem mestu (kolikšen odstotek prebivalcev države živi v glavnem mestu). Izpisane podatke loči s podpičjem, zapisi pa naj bodo urejeni padajoče po deležu prebivalcev v glavnem mestu. Predpostaviš lahko, da ima vsak zapis svoj par in da so argumenti programa vedno podani.

Primer:

Prva datoteka (`drzave.txt`)

```
SLO;Slovenija;2078938
AUT;Avstrija;9003627
ITA;Italija;60461826
```

Druga datoteka (`mesta.txt`)

```
AUT;Dunaj;1691468
ITA;Rim;2318895
SLO;Ljubljana;272220
```

Izpis programa ob klicu `java Naloga34 drzave.txt mesta.txt:`

```
Avstrija;Dunaj;18.8
Slovenija;Ljubljana;13.1
Italija;Rim;3.8
```