

Naloge za prvo srečanje

Bash

1. Znašli smo se v precej čudni situaciji, ko bi radi dostopali do spleta, pa ne moremo. Vaša naloga je, da s trenutnim znanjem ukazov najdete ustrezen IP naslov vašega prenosnika in ga pripnete na konec že znane datoteke *cmdline.txt*. Omenjena besedilna datoteka sme biti spremenjena zgolj pri predelu, ki smo ga v začetku današnjega srečanja dodali sami, saj z napačnim urejanjem lahko povzročite nekaj nevšečnosti pri zagonu mini računalnika. Pri iskanju datoteke *cmdline.txt* je priporočljivo, da se premaknete v korensko mapo datotečnega sistema.

Predel, ki smo ga zapisali v začetku mora biti v formatu:

`ip=169.254.39.71::[IP vašega prenosnika]` (Primer: `ip=169.254.39.71::169.254.53.169`)

2. Bliža se božič in Božiček bo pričel raznašati darila otrokom. Ker je že zelo star in stvari rad pozablja mu moramo pomagati tako, da mu na kratko predstavimo nekaj (statističnih) regij naše države, saj se bo nato spomnil kje je in kam mora iti. Zaradi preglednosti si naredite novo mapo imenovano *vaje*, nato pa mu bomo posvetili ločeno mapo znotraj ravnokar izdelane mape. Z enim ukazom naredite tri podmape *Gorenjska*, *Pomurska*, *Spodnjeposavska* v mapi *bozicek*, ki se mora nahajati v prej ustvarjeni mapi *vaje*. Mape morajo biti strukturirane tako:

| - vaje/

| - bozicek/

| - Gorenjska/

| - Pomurska/

| - Spodnjeposavska/

3. V prejšnji nalogi smo božičku pomagali priti zgolj do treh regij, še vedno pa mu nismo podali vseh ključnih podatkov na podlagi katerih bi se lahko spomnil kam mora iti. Podali mu bomo še podatke o nekaj občinah, ki se nahajajo v določeni regiji. Znotraj mape *bozicek* ustvarite prazno besedilno datoteko *obcine.txt* in jo skopirajte v vsako podmapo, nato pa izbrišite (zgolj iz trenutne mape). V vsako datoteko *obcine.txt* posamezne regije zapišite po 3 občine, ki se nahajajo v določeni regiji (seveda se jih lahko tudi izmislite). Občine morajo biti zapisane vsaka v svoji vrstici brez kakršnihkoli ločil. Primer datoteke *obcine.txt* v mapi regije *Obalno-kraška*:

“ Piran
 Izola
 Koper

”

4. Ker se božiček ne spozna najbolje na tehnologijo, mu bo pri iskanju potrebno pomagati.
- a) Ker smo naše vaje “podarili” novemu uporabniku, se v konzolo najprej prijavite kot “uporabnik” (v programu PuTTY lahko podvojite sejo, da se ne boste potrebovali vsakič znova prijavljati z drugim uporabnikom). Mapo *bozicek*, ki ste jo ustvarili v eni izmed prejšnjih nalog skopirajte v vašo domačo mapo.
 - b) Izpišite vse mape in datoteke, ki se nahajajo v mapi *bozicek* vključno z izhodiščno mapo. Vse mape morajo biti izpisane vsaka v svoji vrsti v formatu absolutne poti. Primer izpisa:
“ /home/uporabnik/bozicek/
 /home/uporabnik/bozicek/Obalno-kraška/
 /home/uporabnik/bozicek/Obalno-kraška/obcine.txt ”
 - c) Za hitrejši dostop do indeksa mape zgornji izpis shranite v datoteko *seznam.txt*, ki naj se nahaja v mapi *bozicek*.
 - d) Naredite seznam vseh občin in jih zapišite v datoteko *seznam_obcin.txt*.
 - e) Seznam občin sedaj uredite (razvrstite) po abecedi in jih ponovno zapišite v datoteko *seznam_obcin.txt* (prepišite prej ustvarjeno datoteko).
 - f) Izpišite vse podmape mape *bozicek*.
5. Zapustite sejo uporabnika “uporabnik” in se vrnite nazaj na sejo uporabnika “pi”. Vaša naloga je, da ustvarite prav svoj uporabniški račun, katerega ime naj bo vaše ime ali priimek ali vaše ime in priimek skupaj (brez presledkov, šumnikov, ostalih posebnih znakov; uporabite le črke angleške abecede). Ne odpirajte še nove seje z ravno ustvarjenim up. rač.
- a) Vašega novega uporabnika dodajte v skupino “skupina”, ki smo jo ustvarili pred kratkim.
 - b) Prijavite se v vaš novi uporabniški račun (podvojite sejo ter se prijavite) in v domači mapi ustvarite mapo z imenom *vaje*. Sejo ugasnite (odjavite se iz uporabniškega računa).
Z naslednjo točko bomo nadaljevali po krajši obrazložitvi nekaterih pomembnih stvari.
 - c) Prijavite se v vaš novi uporabniški račun in se premaknite v mapo */home/* ter ustvarite novo mapo z imenom *skupno*. Ne gre? Uporabite magični ukaz. Pozorno in ☺ glasno ☺ preberite sporočilo, ki vam ga je poslal sistem. To je tisto o čemer smo govorili nedavno.
 - d) Mapi skupno uredite lastništvo tako, da bo vaš novi uporabnik lastnik mape, videli, pisali in izvajali stvari v njej pa bodo lahko vsi v skupini *skupina* in nihče drug.
 - e) Sejo s trenutnim uporabnikom ohranite ter podvojite sejo in se prijavite kot uporabnik “uporabnik”. Iz domače mape izbrišite mapo *bozicek*. Mapo *vaje* “premknite” v prej ustvarjeno skupno mapo.
 - f) Ravno kar “premknjeni” mapi uredite pravice tako, da bodo vanjo lahko pisali tudi uporabniki skupine “skupina”.

- g) Premaknite se na sejo kjer ste prijavljeni z uporabnikom, ki je poimenovan po vašem imenu/priimku in v mapi *skupno* ustvarite datoteko z imenom *povezava.txt* in vanjo prilepite naslov [http:// \[IP\] /python.tar.gz](http://[IP]/python.tar.gz). Ker uporabljate uporabniški račun z magičnimi pravicami, izbrišite še mapo *vaje* iz domače mape uporabnika »uporabnik«. Premaknite se na sejo omenjenega uporabnika in preverite, če je mapa zares izbrisana, nato pa le-to sejo ugasnite.

Z naslednjo točko bomo nadaljevali po krajši obrazložitvi nekaterih pomembnih stvari.

6. V tej nalogi napišite ukaz v eni vrstici, brez, da posredujete prebrane podatke neposredno na katerikoli drug ukaz v novi vrstici. Najprej s prvim ukazom preberite datoteko *povezava.txt*, ki ste jo ustvarili v zadnjem koraku prejšnje naloge in posredujte prebrano kot argument drugemu ukazu, ki omogoča prenos datotek.
7. Po uspešnem zaključku prejšnje naloge bi morali prenesti arhivsko datoteko. V kolikor se to ni zgodilo ponovno rešite prejšnjo nalogo, sicer pa razpakirajte njeno vsebino v “podmapo” *vaje* v vaši domači mapi poleg datoteke *povezava.txt*. Razširjene datoteke bomo potrebovali za nadaljnje delo.

Python

1. Napišite program, ki sprejme 4 vhodne podatke različnega podatkovnega tipa. Prvi podatek naj bo celoštevilski, drugi naj bo podan kot decimalno število, tretji kot niz in četrti kot logična vrednost. Vse podatke izpišite na zaslon v obliki, kjer podčrtaj označuje vrednost:

```
Celo število: ____  
Decimalno število: ____  
Niz: ____  
Logična vrednost: ____
```

Nato pa izpišite podatke še v eni vrstici z začetnimi črkami posameznega podatkovnega tipa, med njimi pa naj bo znak »|« (*alt gr + w*):

```
C: ____ | D: ____ | N: ____ | L: ____
```

2. Napišite program, ki sprejme pet celoštevilskih vrednosti in jih med seboj primerja. Za vsako primerjavo naj izpiše večje in manjše število oziroma enakost le-teh. Primerjave naj prične izvajati po prvem vnesenem številu.
3. Napišite program, ki kot vhodne podatke sprejema števila, dokler ne zapišemo znakov **X** ali **x**. Nad števili naj izvaja vse osnovne aritmetične operacije in šteje koliko vhodnih podatkov smo podali. Po koncu izvajanja naj izpiše rezultate vseh operacij ter število podanih vhodnih podatkov.
4. Napišite program, ki vsebuje funkcije *sestěj*, *odstěj*, *mnozi*, *deli* za prikladne aritmetične operacije nad dvema podanima številoma. Funkcije naj ob klicu rezultat vrnejo v ustreznem podatkovnem tipu.

5. Napišite program, ki vsebuje funkcije iz prejšnje (4.) naloge in kot vhodne podatke sprejema števila. Za vsaki dve vneseni števili uporabnika vpraša po željeni aritmetični operaciji, ki se izvede nad prejetima številoma. Operacije naj bodo označene s števili od 1 naprej, uporabniku pa morate omogočiti izvajanje kompozituma funkcij.

★ Dodatna naloga:

Vse vnesene vrednosti in operacije shranjujte v ustrezne podatkovne strukture ter na koncu izpišite dnevnik dejavnosti uporabnika v programu.

6. Napišite program, ki bo sestavljal slovarje v slovarju znotraj slovarja tako, da bo uporabnik najprej napisal kategorijo jedi, nato naslov jedi in potrebne sestavine za pripravo dodane jedi. Uporabnik lahko katerikoli vnos prekliče tako, da v vnosno polje vpiše znak **X** ali **x**. Po koncu vnašanja podatkov izpišite kategorije jedi, vse naslove jedi in zberite vse sestavine, ki se pojavljajo pri jedeh. Pazite, da se vam podatki znotraj podatkovnih struktur (razen pri sestavinah) ne podvajajo, kar velja tudi pri izpisu (v tem primeru to drži tudi za sestavine).
7. Napišite program za legendarno igro vislice, v kateri je potrebno ugotoviti naključno izbrano besedo. Besede so na voljo v datoteki *stiri.txt* v arhivu, ki ste ga pri eni izmed prejšnjih nalog prenesli in razpakirali. Cilj naloge je izdelati polno igro, ki uporabnika sprašuje po črkah besede, ta pa ima na voljo 5 poskusov (vnosov črke) za vsako besedo. Uporabnik dobi 5 točk za vsako pravilno najdeno besedo ter -10 točk, če preseže število poskusov. Igra se prekine z bližnjico »KeyboardInterrupt«, nakar program izpiše število skupno, pravilno in nepravilno uganjenih besed ter število točk, ki jih je uporabnik dosegel. Uporabniku lahko pomagata tako, da mu razkrijeta eno izmed črk v besedi.
8. Napišite program, ki bo prebral datoteko *besede.txt*, ki ste jo prenesli in razpakirali v eni izmed prejšnjih nalog. Program naj preveri vsako besedo v novi vrstici ter upošteva pogoje v nadaljevanju, nato pa za besede dolge 5, 8, 10 in več kot 12 znakov ustvari nove datoteke *pet.txt*, *osem.txt*, *deset.txt*, *vec.txt* v katere bo zapisoval ustrezne besede. Pogoji za vse besede, ki jih bo zapisoval so, da ne vsebujejo pik, presledkov, oklepajev in zaklepajev ter šumnikov. Program naj besede zapisuje v datoteke glede na dolžino le-teh - torej besede, ki so dolge 5 znakov zapiše v datoteko *pet.txt*, besede dolge 8 znakov zapiše v datoteko *osem.txt* in tako naprej. Dodatna zahteva je, da program zapiše po pet besed, ki se začnejo s posamezno črko abecede (brez šumnikov), hkrati pa naj se črke v besedah ne ponavljajo.

Preden se lotite naloge dobro razmislite o tem katere podatkovne tipe in strukture boste potrebovali. Primarna lastnost programa naj bo, da bo nove datoteke izdelal karseda hitro, zato se držite podanih dodatnih zahtev. Ne pozabite na modularnost programa.

★ Dodatna naloga:

Dopolnite prejšnji program igre vislic tako, da bo v začetku uporabnika vprašal po težavnostni stopnji in nato naložil primerno datoteko v seznam:

- ➔ Enostavno – besede s štirimi črkami (datoteka *stiri.txt*)
- ➔ Zahtevno – besede s petimi črkami (datoteka *pet.txt*)
- ➔ Težko – besede z osmimi črkami (datoteka *osem.txt*)
- ➔ Težje – besede z desetimi črkami (datoteka *deset.txt*)
- ➔ Najtežje – besede z več kot dvanajst črkami (datoteka *vec.txt*)

Dodajte tudi posebno možnost za napredovanje – ko uporabnik ugame 5 enostavnih besed, mu samodejno ponudite besede iz naslednje težavnostne stopnje. Dobro razmislite o tem, kdaj bo vaš program naložil besede iz datotek tako, da uporabnik ne bo predolgo čakal.