

Osnove programiranja 2

Naloge laboratorijskih vaj

za študijsko leto 2006/2007

Navodila o opravljanju laboratorijskih vaj so objavljena na spletnih straneh predmeta.

Naloge iz **C** morate zagovoriti v prvi polovici semestra, to je do **20. aprila 2007**.

Naloge iz **Java** morate zagovoriti najkasneje do konca semestra, to je do **31. maja 2007**.

Naloge iz programskega jezika C (do 20. aprila 2007)

1. V programskem jeziku C napišite program, ki v datoteki prešteje vse črke angleške abecede (a - z in A - Z) in na koncu izpiše, kolikokrat se vsaka črka v datoteki pojavi. Pri tem ne ločimo velikih in malih črk. Ime datoteke programu podamo kot parameter ukazne vrstice. (1 točka)
2. V programskem jeziku C napišite program, ki primerja vsebini dveh datotek in izpiše vse tiste vrstice, ki se razlikujejo, skupaj z ustrežno številko vrstice. Imeni obeh datotek sta podani kot parametra ukazne vrstice, dolžina vrstice pa je omejena na 80 znakov. (1 točka)
3. V programskem jeziku C napišite program, ki reši problem Hanoiskih stolpov in izpiše potek reševanja.
Pri problemu Hanoiskih stolpov je potrebno prestaviti n plošč različnih velikosti iz stolpa a na stolp c s pomočjo stolpa b , pri tem pa je potrebno paziti, da večja plošča nikoli ne pride na manjšo. Vedno lahko prestavimo le eno ploščico naenkrat in vedno le tisto, ki je na vrhu.

Primer za $n = 3$ dobimo naslednji izpis:

```
a -> c
a -> b
c -> b
a -> c
b -> a
b -> c
a -> c
```

Nasvet: Pomagajte si z rekurzijo.

(1 točka)

4. V programskem jeziku C napišite program, ki preveri, ali je podana beseda palindrom. Palindrom je niz, ki se enako bere naprej in nazaj (primer: ata, abba, cepec, radar, Lagerregal). Pri tem morate funkcijo za preverjanje palindromov implementirati rekurzivno. (1 točka)

5. V programskem jeziku C napišite program, ki v podanem nizu znakov poišče najdaljši podniz, ki je palindrom. Palindrom je niz, ki se enako bere naprej in nazaj (primer: ata, abba, cepec, radar, Lagerregal). Primer: najdaljši palindrom v nizu

"to je niz zz **lagerregal**ia plindromom"

je "lagerregal" (označen krepko v nizu), kar tudi izpiše program. Sicer podani niz vsebuje več palindromov.

(1 točka)

6. V programskem jeziku C napišite program, ki sortira linearni seznam. Seznam mora biti implementiran kot kazalčni seznam in naj hrani nize znakov, soritranje pa je naraščajoče po (angleški) abecedi. Preverite pravilnost delovanja programa tako, da zgradite poljuben seznam (ki vsebuje vsaj 10 elementov), ga sortirate in izpišete.

Za sortiranje lahko uporabite poljuben algoritem, a ga morate implementirati sami.

(1 točka)

7. V programskem jeziku C napišite program, ki z linearnim seznamom ali z dinamično ustvarjeno tabelo implementira krožni medpomnilnik (*buffer*). Program naj vsebuje funkcijo za inicializacijo medpomnilnika (podamo ji velikost medpomnilnika) ter funkcije za dodajanje, odzemanje, štetje in izpis elementov v medpomnilniku.

(1 točka)

8. Nize, v katerih se neka črka večkrat ponovi, lahko zapišemo krajše tako, da za črko pripišemo število ponovitev:

aa	zapišemo kot	a2
aaaAbbbb	zapišemo kot	a3Ab5
aaaBBBBBBBBBBBc	zapišemo kot	a3B12c

Tako stisnjena beseda ni nikoli daljša od originala, saj število zaporednih nastopanj vsake črke dodamo le v primeru, ko se črka dejansko ponavlja.

V programskem jeziku C napišite program, ki omogoča stiskanje besed. Program naj bo omejen na besede, ki so sestavljene samo iz črk angleške abecede (a - z in A - Z).

Upoštevajte tudi, da je število ponovitev lahko tudi večmestno število (poljubno veliko, tudi večje od sto). Besedo, ki jo stiskamo, naj program prebere s standardnega vhoda in stisnjeno besedo izpiše na standardni izhod.

(1 točka)

Program dopolnite s funkcijo za razširjanje besed, ki so stisnjene na zgoraj opisani način.

(+1 točka)

9. V programskem jeziku C napišite program, ki bo deloval kot RPN (Reverse Polish Notation) kalkulator. Tak kalkulator sprejme za vsako operacijo najprej operande (v splošnem realna števila) in nato prebere še operator, ki nadomesti operande z rezultatom operacije. Rezultat lahko nastopa kot operand neke naslednje operacije. Vaš kalkulator naj omogoča računanje z operatorji: + (seštevanje), - (odštevanje), * (množenje), / (deljenje), -- (unarni minus), ** (kvadrat) ter // (kvadratni koren).

Primer izračuna izraza $(2+3)*(4+5)$:

```
2 <ret>
2
3 <ret>
3
+ <ret>
5
4 <ret>
4
5 <ret>
5
+ <ret>
9
* <ret>
45
```

Nasvet: Pomagajte si s skladom.

(1 točka), če je velikost izraza omejena in
(2 točki), če velikost izraza ni omejena.

10. Razpršena matrika je matrika velikosti $m \times n$, ki ima večino elementov ničelnih. Ker je hranjenje velikega števila ničelnih elementov v dvodimenzionalni tabeli potratno, je primerneje razpršeno matriko zapisati v obliki kazalčnega seznama neničelnih elementov. Pri tem vsak element seznama poleg vrednosti vsebuje še oba indeksa tega elementa v matriki in kazalec na naslednji element:

```
struct element
{
    int mind, nind;           /*indeksa elementa v matriki*/
    int vrednost;            /*vrednost elementa*/
    struct element *nasl;    /*kazalec na naslednji element*/
}
```

V programskem jeziku C napišite program za delo z razpršenimi matrikami. Program naj omogoča branje matrike v tabelarični obliki z vhoda (najprej velikost matrike, potem vsi elementi matrike), hranjenje matrike v obliki kazalčnega seznama, izpis s kazalčnim seznamom predstavljene matrike na zaslon (standardni izhod) ter funkcijo za računanje vsote dveh matrik enakih dimenzij, ki sta zapisani v obliki kazalčnega seznama. Matrika naj bo deklarirana kot kazalec na zapis, ki vsebuje podatka o obeh dimenzijah matrike ter kazalec na začetek seznama z elementi matrike.

(2 točki)

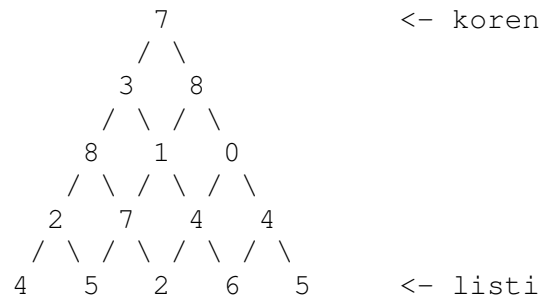
Program dopolnite s funkcijo za računanje produkta dveh matrik velikosti $m \times k$ in $k \times n$ (velikost produkta je $m \times n$).

(+1 točka)

11. V programskem jeziku C napišite program, ki prebere datoteko in poišče ter izpiše 10 najbolj pogosto uporabljenih besed. Besede so lahko poljubno dolge, ločilo med besedami pa so prazni prostori (angl. *whitespace*), kot so presledek, tabulator, znak za novo vrstico ... Program naj datoteko, katere ime je podano kot argument ukazne vrstice, prebere samo enkrat (zaporedno branje). Za shranjevanje podatkov o prebranih besedah uporabite linearni

kazalčni seznam.
(3 točke)

12. Na sliki je prikazan trikotnik med seboj povezanih števil. V programskem jeziku C napišite program, ki bo izračunal največjo vsoto števil na poti od korena do končnega lista. Na tej poti se lahko pomikamo samo po povezavah in v smeri navzdol. Predpostavimo, da drevo ne bo imelo več kot 20 končnih listov, vrednosti pa so cela števila od vključno 0 do vključno 9.



Program naj prebere trikotnik števil preko standardnega vhoda. V prvi vrstici vhoda je število končnih listov, sledijo vrstice z vrednostmi števil v drevesu. Za prikazan zgled bi bila vsebina na vhodu:

```

5
7
3 8
8 1 0
2 7 4 4
4 5 2 6 5
```

Program naj na standardni izhod izpiše vrednost največje vsote ter pot od korena do lista za to vsoto. V našem primeru:

```

30
      7
     /
    3  8
   /
  8  1  0
   \
  2  7  4  4
   /
  4  5  2  6  5
```

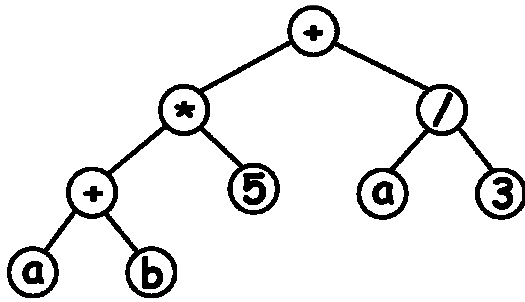
(3 točke)

13. V programskem jeziku C napišite program `parse`, ki iz ukazne vrstice prebere aritmetični izraz v *infixni* obliki in ga izpiše v *prefiksni* in *postfiksni* obliki. Pri postopku si lahko pomagata z binarnim drevesom, ki ga predstavite s kazalci.

Aritmetični izrazi lahko vsebujejo:

- operatorje: **+**, **-**, ***** in **/** z običajnimi prioritetami,
- oklepaje: **(in)** ter
- operande: **števila** ali **konstante** (črke od a do z).

Primer: ukazu `parse (a+b)*5+a/3` pripada drevo



in izpisa:

```
+ * + a b 5 / a 3
a b + 5 * a 3 / +
```

Pazite na pravilne prioritete operatorjev!
(4 točke)

14. Sami si zamislite problem in ga rešite s programom v programskem jeziku C. Z asistentom se pogovorite o primernosti in težavnosti problema ter pripadajočem številu točk.
(do 5 točk)

Naloge iz programskega jezika Java (do 31. maja 2007)

NAVODILO: Pri reševanju nalog iz programskega jezika Java ne smete uporabljati razreda `BranjePodatkov`, ki ste ga spoznali pri Osnovah programiranja 1. Pri delu z datotekami in standardnim vhom/izhodom (branje, pisanje, ...) uporabljajte Javine knjižnice.

1. V programskem jeziku Java napišite aplet za spletno menjalnico, ki omogoča pretvorbo iz petih različnih tujih valut (CHF, JPY, GBP, HRK in USD) v evre (EUR) in obratno. Pretvorba naj se izvaja s pomočjo tabele menjalnih tečajev, ki jo vzpostavite s pomočjo seznama vrednosti, na primer:

```
double[] tecaji = {1.6267, 157.78, 0.67115, 7.3557, 1.3137};
```

Aplet naj omogoča tudi vpis novih vrednosti menjalnih tečajev. Osrednji del okna naj zajema vnosno polje za znesek, ki ga želimo pretvoriti, oznako za izpis izračunane vrednosti v drugi valuti ter deset gumbov, ki omogočajo vse mogoče pretvorbe med EUR in tujimi valutami. Po potrebi lahko aplet vsebuje tudi dodatne oznake, vnosna polja in gumbе (kjer je smiselno, uporabite polje elementov).

(1 točka)

2. V programskem jeziku Java napišite program, ki omogoča pretvarjanje med arabskimi in rimskimi števili. Za predstavitev rimskih števil uporabite naslednje znake: I za 1, V za 5, X za 10, L za 50, C za 100, D za 500 in M za 1000. Pri pretvorbi se zaradi enostavnosti omejite na števila do 5000.

Program naj pretvarja iz arabskih v rimska števila.

Primer:

```
java pretvori 1990
izpiše MCMXC
```

```
java pretvori 2007
izpiše MMVII
```

(1 točka)

3. V programskem jeziku Java napišite program, ki omogoča pretvarjanje med arabskimi in rimskimi števili. Za predstavitev rimskih števil uporabite naslednje znake: I za 1, V za 5, X za 10, L za 50, C za 100, D za 500 in M za 1000. Pri pretvorbi se zaradi enostavnosti omejite na števila do 5000.

Program naj pretvarja iz rimskih v arabska števila.

Primer:

```
java pretvori MCMXC
izpiše 1990
```

```
java pretvori MMVII
izpiše 2007
```

(1 točka)

4. Napišite program v programskem jeziku Java, ki implementira enosmerni seznam (razred `List`). Elementi seznama hranijo cela števila. Seznam je lahko tipa vrsta, sklad ali urejen seznam. Za dodajanje elementa v seznam uporabite metodo `storeit()`, za odvzemanje elementa iz seznama pa metodo `retrieveit()`. Osnovni razred `List` naj bo abstrakten in naj dokončno ne definira metod teh dveh operacij. Obe metodi naj bosta določeni v izvedenih razredih `Queue`, `Stack` in `SortList`. V seznamu tipa vrsta naj se elementi dodajajo na koncu in odvezemajo na začetku, v seznamu tipa sklad naj se elementi dodajajo in odvezemajo na koncu, v seznamu tipa urejen seznam pa naj se elementi dodajajo na ustrezno mesto glede na naraščajoči vrstni red shranjenih celih števil in odvezemajo na začetku seznama. Program naj generira vse tri sezname in ilustrira dinamično izbiranje prekritih metod. Tip seznama, celo število in operacija naj se izbirajo interaktivno.

(1 točka)

5. V programskem jeziku Java napišite program, ki iz podane datoteke z Javanskim programom zbriše vse komentarje (vse tri vrste komentarjev, ki jih pozna Java). Tako očiščeno kodo zapiše v izhodno datoteko, če je le-ta podana, sicer pa jo zapiše nazaj v isto datoteko.

(1 točka)

6. V programskem jeziku Java napišite program, ki bo podano besedo prevedel v Morseovo abecedo, pri tem naj se med črkami Morseove abecede nahaja presledek. Definirajte razred `Morse`, ki bo vseboval vse črke abecede in pripadajočo šifro. Ta razred potem uporabite v

razredu Kodirnik, kjer bo šifriranje potekalo. Tabela črk Morseove abecede najdete na koncu tega dokumenta.

(1 točka)

7. V programskem jeziku Java sestavite program za izris preprostega grafa. Enačba po kateri se naj izrisuje graf je naslednja:

$$f(x) = a * \sin(b * x + c) + d * \cos(e * x + f)$$

Vse parametre a , b , c , d , e in f funkcije $f(x)$ naj bo mogoče spreminjati med delovanjem programa. Za urejanje parametrov in izris grafa izdelajte primeren uporabniški vmesnik, za katerega uporabite ustrezen razporejevalnik (*layout manager*). Poskrbite za osveževanje grafa ob spremembi parametrov. Ne pozabite na izris osi x in y . Poskrbite tudi za primerno območje izrisovanja grafa.

Namig: upoštevajte amplitudi kotnih funkcij.

(1 točka)

8. V programskem jeziku Java napišite grafični program, ki deluje kot kalkulator. Omogoča naj izračun osnovnih štirih operacij (+, -, x, /), kvadrata in kvadratnega korena. Poleg tega naj kalkulator omogoča tudi računanje štirih funkcij: $\sin(x)$, $\cos(x)$, e^x ter $\ln(x)$. Ne pozabite tudi na tipke za izbris vrednosti, izbris zadnje vnešene vrednosti, vnos decimalne vejice ter vnos negativnega števila. Vse elemente ustrezno razporedite v oknu, da bo kalkulator enostaven za uporabo. Pri računanju morate upoštevati prioriteto operatorjev. Nasvet: Delajte s skladom.

(1 točka)

Kalkulator naj omogoča tudi izris vseh štirih funkcij, ki jih računa.

(+1 točka)

9. V programskem jeziku Java napišite program `multigrep`, ki deluje podobno kot ukaz `grep` z več besedami. Program naj prejme ime datoteke kot prvi argument, ostali argumenti pa so besede, ki jih iščemo. Program izpiše vse tiste vrstice iz podane datoteke, ki vsebujejo eno ali več podanih besed. Primer klica programa:

```
java multigrep dat.txt prvabeseda drugabeseda tretjabeseda
```

(1 točka)

Programu dodajte tudi stikalo `-i` (*ignore case*), ki omogoča iskanje besed ne glede na velike oz. male črke, stikalo `-n` (*number*), ki pred vsako izpisano vrstico izpiše tudi njeno zaporedno številko, ter stikalo `-c` (*count*), ki vrstic ne izpisuje, temveč jih le prešteje in izpiše skupno število vrstic, ki zadostujejo pogoju. Vsa stikala so opcijska in se lahko uporabljajo tudi hkrati.

(+1 točka)

10. Napišite program v programskem jeziku Java, ki prebere vsebino HTML datoteke in izpiše URL naslove vseh povezav (linkov), navedenih v tej datoteki. HTML datoteka je podana z URL naslovom.

Program naj omogoča vpis spletnega naslova datoteke (URL) v vnosno polje v oknu, izpis povezav pa naj bo v ustreznem elementu v istem oknu.

Pri tem si pomagajte z objektoma tipa `java.net.URL` in `java.io.BufferedReader`.

(1 točka)

Program naj v oknu prikaže vse slike, ki so v podani datoteki. Slike so v HTML datoteki navedene z oznako "img".

(+1 točka)

11. V datoteki imamo shranjene podatke o naših glasbenih datotekah. Podatki naj obsegajo izvajalca, naslov pesmi, naslov albuma in leto izida. V programskem jeziku Java napišite program (v grafičnem okolju) za pregledovanje in spreminjanje podatkov, za dodajanje novih pesmi in brisanje obstoječih. Vsaka glasbena datoteka naj bo predstavljena z enim objektom (definirajte ustrezen razred), datoteka pa naj hrani cele objekte (to ni tekstovna datoteka).

(2 točki)

12. V programskem jeziku Java napišite program, ki bo omogočal prostoročno risanje z miško. Program naj omogoča izbiro različnih barv pisala. Uporabnik naj riše tako, da pritisne na miškin gumb in miško vleče po risalni površini, na mestu kurzorja pa se sproti izrisujejo točke v izbrani barvi. Ko gumb spusti, se risanje konča. Pri izdelavi bodite pozorni na osveževanje risalne površine oz. zaslona in uporabljajte tehniko dvojnega pomnjenja (*double buffering* - risanje naj poteka v bitno sliko, ta pa naj se izrisuje na zaslon).

(2 točki)

Programu dodajte opciji, ki omogočita nalaganje in shranjevanje narisanih slik iz/v datoteko na disku. Pri tem bodite pozorni na možne izjeme (datoteka ne obstaja, je v napačnem formatu, je ne morem shraniti, ...) in izpišite primerna obvestila o napakah.

(+1 točka)

Programu dodajte opcijo, ki razveljavi zadnjo narisano potezo (*undo*). Globina razveljavljanja naj bo poljubna: uporabnik lahko torej z večimi zaporednimi izbirami te opcije razveljavi zadnjih nekaj narisanih potez.

(+1 točka)

Programu dodajte novo orodje, ki služi zapolnjevanju (barvanju) delov risalne površine (*free fill*). Ob izbiri orodja lahko uporabnik z miško klikne na poljuben del risalne površine, ki se pri tem pobarva z izbrano barvo do roba; rob je določen z barvo, ki je drugačna od barve točke, na katero je uporabnik kliknil.

(+2 točki)

Programu dodajte izpis trenutnega časa (`ure:minute:sekunde`). Za sprotno izpisovanje trenutnega časa naj poskrbi nova nit.

(+1 točka)

13. Napišite program v programskem jeziku Java, ki simulira premikanje žogic v oknu. Ob zagonu programa naj bo v oknu ena žogica. Žogica naj ima smer in hitrost premikanja. Ko pride žogica do ovire (rob okna, druga žogica), se od ovire odbije. Program naj omogoča simulacijo poljubnega števila žogic (premikanje vsake žogice simulira svoja nit). Program naj omogoča dodajanje novih žogic v okno. Za vsako novo dodano žogico skrbi nit, ki se ustvari na novo. Pri dodajanju nove žogice se njeni začetni parametri (začetni položaj, smer in hitrost) nastavijo naključno.

(3 točke)

14. Sami si zamislite problem in ga rešite s programom v programskem jeziku Java. Z asistentom se pogovorite o primernosti in težavnosti problema ter pripadajočem številu točk. (do 5 točk)

MORSEOVA abeceda

A .-
B -...
C -.-.
D -..
E .
F ..-.
G --.
H
I ..
J .---
K -.-
L .-...
M --
N -.
O ---
P .-.-.
Q --.-
R .-..
S ...
T -
U ..-
V ...-
W .--
X -...-
Y -.-.-
Z --...
1 .-----
2 ..----
3 ...--
4-
5
6 -.....
7 --.....
8 ----..
9 -----.
0 -----

Za morebitne pripombe in vprašanja v zvezi z nalogami se obrnite na asistente pri vajah.