

Osnove programiranja 2 - Naloge za šolsko leto 2001/2002

Navodilo za reševanje in zagovor nalog

Naloge so razdeljene po težavnosti na lahke, srednje težke in težke naloge. Vse naloge so točkovane glede na težavnost od ene točke (lahke naloge) do štirih ali več točk (težke naloge). Število točk, ki jih prinaša posamezna naloga, je napisano v oklepajih pri besedilu naloge.

Za uspešno opravljene laboratorijske vaje, ki so pogoj za pisni izpit, morate rešiti in uspešno zagovoriti toliko nalog, da zberete skupaj vsaj 13 točk.

Uspešno zagovorjene laboratorijske vaje se ocenjujejo (ocene od 6 do 10), na oceno pa vpliva predvsem kakovost rešitev nalog in izbira nalog (težje naloge prinesejo boljšo oceno). Vaje lahko sicer opravite tudi z zagovori samih lahkih nalog, a je v tem primeru, kljub kakovostnim in originalnim rešitvam nalog, ocena največ 8.

Naloge mora reševati vsak sam, skupinsko delo ni dovoljeno. Naloge lahko delate tudi doma, vendar jih morate zagovarjati na laboratorijskih vajah na UNIX sistemih.

Vse naloge morate zagovoriti do konca predzadnjega tedna v maju (do 24. maja 2002).

Naloge

1. V programskem jeziku C napišite program, ki izračuna vrednost polinoma v podani točki. Program naj najprej zahteva vpis stopnje polinoma, zatem naj prebere koeficiente posameznih členov polinoma ter točko, v kateri računa vrednost polinoma. Na koncu naj program na zaslon izpiše rezultat.
(1 točka)
2. V programskem jeziku C napišite program, ki preveri, ali je podano število praštevilo. Program naj prebere število in izpiše, ali vnešeno število je oz. ni praštevilo.
(1 točka)
3. V programskem jeziku C napišite program, ki iz izvorne kode programa, pisanega v jeziku C, izloči komentarje. Komentarje označuje niz // ali pa par /* in */.

Primer komentarjev:

```
// to je komentar, do konca vrstice  
/* tudi to je komentar, ki pa lahko obsega vec vrstic */  
(1 točka)
```

4. V programskem jeziku C napišite program sestej, ki prebere dva niza iz ukazne vrstice. Niza imata naslednji pomen: prvi niz predstavlja dnevni čas (npr. 13:45 predstavlja 13 ur in 45 minut), drugi pa pretečeni čas v enakem formatu. Program mora oba niza sešteti, dobljeni rezultat pa izpisati na standarden izhod.

Primer:

```
sestej 13:45 0:35
izpiše
ura je: 14:20
Upoštevati je treba tudi preskok na naslednji dan:
sestej 22:30 2:35
ura je: 1:05
(1 točka)
```

5. V programskem jeziku C napišite program, ki izpiše 10 najdaljših besed v datoteki. Ime datoteke je podano z vhodnim parametrom programa.
(1 točka)
6. V programskem jeziku C napišite program, ki primerja vsebini dveh datotek in izpiše vse tiste vrstice, ki se razlikujejo, skupaj z ustrezno številko vrstice. Imeni obeh datotek sta podani kot parametra ukazne vrstice.
(1 točka)
7. V programskem jeziku C napišite program, ki v datoteki prešteje vse cifre (znake med 0 in 9) in na koncu izpiše, kolikokrat se vsaka cifra v datoteki pojavi. Ime datoteke programu podamo kot parameter ukazne vrstice.
(1 točka)
8. V programskem jeziku C napišite program, ki iz linearnega seznama izloči vse duplikate elementov. Program naj najprej generira linearni seznam, ki vsebuje naključna števila med ena in deset, nato pa naj iz seznama izloči vse elemente, ki se pojavijo več kot enkrat ter dobljeni seznam izpiše na zaslon.
(1 točka)
9. V programskem jeziku C napišite program, ki omogoča osnovne operacije nad prioriteto vrsto (urejni enosmerni kazalčni seznam). Elementi seznama naj hranijo cela števila, ki določajo tudi prioriteto elementa. Napišite funkcije za inicializacijo vrste, dodajanje novega elementa v vrsto, izločanje elementa iz vrste, izpis vseh elementov v vrsti in izpis števila elementov v vrsti.
(1 točka)

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 mideks, nideks;           /*indeksa elementa v matriki*/
    int vrednost;                /*vrednost elementa*/
    struct element *naslednji;    /*kazalec na naslednji element*/
}
```

11. V programskem jeziku C napišite program za delo z razpršenimi matrikami. Program naj omogoča branje matrike v tabelarični obliki z vhoda, hranjenje matrike v obliki kazalčnega seznama, izpis s kazalčnim seznamom predstavljene matrike na zaslon (standatdi 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.

(1 točka)

12. V programskem jeziku C napišite program, ki v podanem nizu poišče najdaljši palindrom. Palindrom je niz, ki se enako bere naprej in nazaj (primer: abba, cepec).

(1 točka)

13. V programskem jeziku C napišite program, ki bere datoteko in vsako vrstico izpiše v obratnem vrstnem redu.

Primer vhoda:

Napisite program, ki bere datoteko in vsako vrstico izpise v obratnem vrstnem redu.

Primer vhoda in njegovega izhoda:

in njegovega ustreznega izhoda:

mentarbo v esipzi ocitsrv okasv ni oketotad ereb ik ,margorp
etisipaN

.uder mentsrv

:adohzi agevoejn ni adohv remirP

(1 točka), če program realizirate s shranjevanjem prebranih znakov, oziroma

(2 točki), če program realizirate brez shranjevanja prebranih znakov.

Program, ki omogoča delo z razpršenimi matrikami (naloga 10), dopolnite s funkcijo za računanje produkta dveh matrik velikosti $m \times k$ in $k \times n$ (velikost produkta je $m \times n$).

(2 točki)

14. V programskem jeziku C napišite program isci, ki v datoteki poišče določen vzorec. Program naj izpiše vse tiste vrstice, v katerih se nahaja iskani vzorec. Iskani vzorec podamo v ukazni vrstici v narekovajih, lahko pa uporabimo tudi opcijo -i, ki ne razlikuje velikih in malih črk, ali opcijo -c, ki samo izpiše številke tistih vrstic, v katerih se iskani vzorec nahaja. Ime datoteke podamo v ukazni vrstici. Upoštevajte tudi možnost kombiniranja stikal (npr. -i -c ali -ic).

(2 točki)

15. V programskem jeziku C napišite program, ki na zaslon izpiše prvih n vrstic Pascalovega trikotnika. Število n je podano kot vhodni parameter. Primer izpisa za $n=6$ prikazuje spodnja slika.

```
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
```

Navodilo za izgradnjo trikotnika: število v trikotniku dobimo tako, da seštejemo števili, ki sta napisani levo in desno v prejšnji vrstici. Na robu je vedno 1. Primer (glej sliko): $5 = 1 + 4$ ali $6 = 3 + 3$.

(2 točki)

16. V programskem jeziku C napišite program, ki meri čas (štoparica). Program naj omogoča naslednje možnosti: začetek merjenja, začasna ustavitev merjenja, nadaljevanje merjenja in konec merjenja. Štoparica naj čas prikazuje v sekundah, minutah in urah. Uporabnik naj ima možnost vklopa in izklopa prikaza stotink.
Namig: čas lahko preberete s funkcijo `gettimeofday`.
(2 točki)

17. V programskem jeziku C napišite program, ki nariše graf funkcije na tekstovni zaslon (80x25). Osnovna varianta naj podpira risanje polinomov in racionalnih funkcij. Uporabnik naj poleg koeficientov funkcije vpiše tudi območje risanja (`xMin`, `xMax` ter `yMin`, `yMax`).
(2 točki)

18. V programskem jeziku C napišite program, ki iz datoteke, podane kot parameter ukazne vrstice, prebere do 1000 števil v tabelo, jih sortira in z metodo bisekcije poišče poljubno število, ki smo ga tudi podali v ukazni vrstici. Program naj na standardni izhod izpiše indeks tega števila v urejeni tabeli.

Pri metodi bisekcije najprej poiščemo vrednosti v krajnih točkah ter na sredini med obema točkama. Če je iskana vrednost večja od vrednosti na sredini ponovimo postopek za srednjo in večjo krajno točko, če pa je vrednost manjša, postopek ponovimo za manjšo krajno točko in srednjo točko. Postopek ponavljamo dokler ne najdemo iskanega elementa ali pa se nam krajni točki ne stakneta.
(2 točki)

19. V programskem jeziku C napišite program, ki prebere datoteko in poišče ter izpiše 10 najbolj pogosto uporabljenih besed. Program naj datoteko prebere samo enkrat. Za shranjevanje podatkov o prebranih besedah uporabite linearni kazalčni seznam.
(2 točki)

20. 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 in izpis elementov v medpomnilniku.
(2 točki)

21. V programskem jeziku C napišite program, ki omogoča osnovne operacije nad urejenim dvosmernim linearnim kazalčnim seznamom. Elementi seznama naj hranijo cela števila. Napišite funkcije za inicializacijo seznama, dodajanje novega elementa v seznam, odzemanje elementa iz seznama, iskanje elementa v seznamu in izpis vseh elementov seznama.
(2 točka)

22. V programskem jeziku C napišite program, ki omogoča osnovne operacije nad binarnim drevesom. Elementi drevesa naj hranijo cela števila in naj sestavljajo urejeno binarno drevo. To pomeni, da ima vsak element v drevesu največ dva sinova, vsi elementi levega poddrevesa so manjši od očeta, vsi elementi desnega poddrevesa pa so večji ali enaki očetu.
Napišite funkcije za izgradnjo drevesa iz danega niza števil, vstavljanje elementa v drevo, izločanje elementa iz drevesa in izpis elementov drevesa (števil) po velikosti.
(2 točki)

23. V programskem jeziku C napišite program, ki izpiše zadnjih n vrstic datoteke, ki jo podamo v ukazni vrstici. Privzeta vrednost n , kadar ta argument ukazne vrstice ni podan, naj bo 7. Pri tem upoštevajte, da je datoteka poljubno dolga in tudi dolžine vrstic so poljubne. Program naj datoteko prebere le enkrat.

Nasvet: pomnilnik za shranjevanje zadnjih vrstic datoteke alocirajte dinamično.

(3 točke)

24. V programskem jeziku C napišite program, ki poravnava vsebino tekstovne datoteke. Ime datoteke in število znakov za poravnavo n (npr. 40) sta podana v ukazni vrstici. Program naj vse vrstice poravnava na levi in desni rob (justify) tako, da je v vsaki vrstici natanko n znakov. Če je vrstica krajša od n znakov, jo poskuša najprej zapolniti z besedami iz naslednje vrstice, potem pa med besede vrstice enakomerno vstavi ustrezno število presledkov. Nov odstavek v datoteki je označen s prazno vrstico.

Na primer, vsebino datoteke:

Naloge so razdeljene po težavnosti na lahke, srednje težke in težke naloge. Vse naloge so točkovane glede na težavnost od ene točke (lahke naloge) do štirih ali več točk (težke naloge). Število točk, ki jih prinaša posamezna naloga, je napisano v oklepajih pri besedilu naloge.

Za uspešno opravljene laboratorijske vaje, ki so pogoj za pisni izpit, morate rešiti in uspešno zagovoriti toliko nalog, da zberete skupaj vsaj 13 točk. Uspešno zagovorjene laboratorijske vaje se ocenjujejo (ocene od 6 do 10), na oceno pa vpliva predvsem kakovost rešitev nalog in izbira nalog (težje naloge prinesejo boljšo oceno). Vaje lahko sicer opravite tudi z zagovori samih lahkih nalog, a je v tem primeru, kljub kakovostnim in originalnim rešitvam nalog, ocena največ 8.

Pri $n = 40$ program preoblikuje v:

Naloge so razdeljene po težavnosti na lahke, srednje težke in težke naloge. Vse naloge so točkovane glede na težavnost od ene točke (lahke naloge) do štirih ali več točk (težke naloge). Število točk, ki jih prinaša posamezna naloga, je napisano v oklepajih pri besedilu naloge. Za uspešno opravljene laboratorijske vaje, ki so pogoj za pisni izpit, morate rešiti in uspešno zagovoriti toliko nalog, da zberete skupaj vsaj 13 točk. Uspešno zagovorjene laboratorijske vaje se ocenjujejo (ocene od 6 do 10), na oceno pa vpliva predvsem kakovost rešitev nalog in izbira nalog (težje naloge prinesejo boljšo oceno). Vaje lahko sicer opravite tudi z zagovori samih lahkih nalog, a je v tem primeru, kljub kakovostnim in originalnim rešitvam nalog, ocena največ 8.

(3 točke)

25. V programskem jeziku C napišite program, ki poišče vse rešitve (92) za problem 8 dam in jih izpiše na zaslon.

Pri problemu 8 dam je potrebno na šahovnico (velikosti 8x8) postaviti 8 dam tako, da se medsebojno ne napadajo. Dama, ki stoji na polju 2,2 napada vsa navpična polja (2,1; 2,3; ... 2,8;), vsa vodoravna polja (1,2; 3,2; ... 8,2) ter polja na obeh diagonalah (1,1; 3,3; ... 8,8; in 3,1; 1,3;).

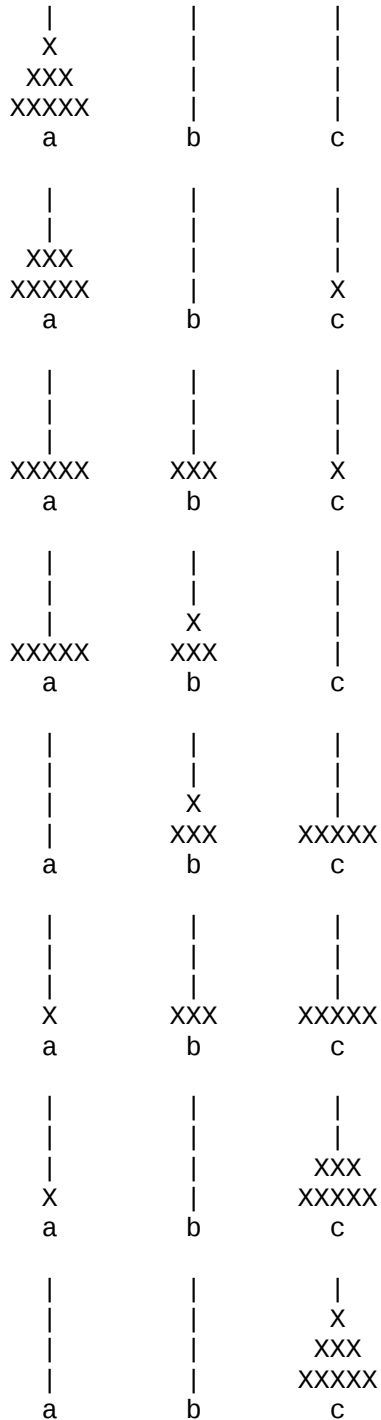
(3 točke)

Rešitev problema 8 dam (naloga 25) dopolnite tako, da iz dobljenih vseh rešitev (92) poiščete vse osnovne rešitve (12), iz katerih je možno izpeljati vse ostale rešitve s pomočjo rotacije (za 90, 180 in 270 stopinj) in zrcaljenja. Osnovne rešitve izpišite na zaslon.

(1 točka)

26. V programskem jeziku C napišite program, ki reši problem Hanoiskih stolpov in grafično prikaže 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.

Primer za $n=3$:

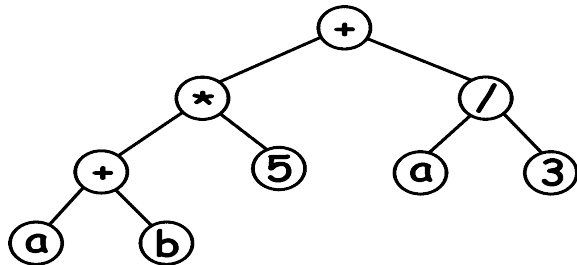


(4 točke)

27. V programskem jeziku C napišite program parse, ki iz ukazne vrstice prebere aritmetični izraz v infiksni 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: ukaz parse (a+b)*5+a/3 pripada drevo



in izpisa: + * + a b 5 / a 3
 a b + 5 * a 3 / +

(4 točke)

28. V programskem jeziku C napišite program za igro štiri v vrsto, ki je definirana na naslednji način: igra se na matriki celic 7 x 7, ki nam predstavlja 7 stolpcev višine 7; igrata jo dva igralca, ki izmenično spuščata (vsak) svoje žetone v stolpce; žetoni v stolpcu vedno padejo na dno stolpca, ki se z nalaganjem postopoma viša; zmaga igralec, ki uspe prvi sestaviti štiri svoje žetone v vrsto, stolpec ali diagonalo.

Program naj omogoča interaktivno igranje proti računalniku, pri čemer naj se računalnik obnaša čimbolj "inteligentno" (samo naključno izbiranje stolpca ni primerno).
 (2 - 4 točke)

Sami si zamislite problem in ga rešite s programom v programskem jeziku C. Z asistentom se pogovorite o primernosti in težavnosti problema (o ustreznem številu točk).
 (1 - 10 točk)