

PiPI VJEŽBA ZI Dodatni problemi za samostalno vježbanje

Grupa 8, Z. Š., 2008.

Napomene

- Namjena ove vježbe je priprema za Završni ispit
- Temeljita priprema pretpostavlja znanje i razumijevanje predavanja (teorija i primjeri)
- Zadane probleme **pokušati riješiti samostalno i tek potom**, ukoliko postoji, pogledati rješenje
- Primjeri su uglavnom poredani po složenosti.

1. Problemi

1.1. Vlastita funkcija umjesto strcpy

Napisati vlastito rješenje funkcije strcpy, ali koja vraća broj kopiranih znakova.

Rješenje: **strcpyMoja.c**

1.2. Vlastita funkcija za konverziju niza znakova u int

Napisati vlastito rješenje funkcije koja niz znakova pretvara u cijeli broj. Pretpostaviti da se niz znakova sastoji samo od znamenaka i da nema predznak.

Dodatna vježba može biti za niz znakova koji ima praznina i predznak te na kraju znakove koji nisu znamenke te se ignoriraju.

Rješenje: **atoiMoja.c i atoiMoja2.c**

1.3. Probiranje znakova

Napisati funkciju koja iz niza znakova izbacuje znak predan kao drugi argument. Funkcija vraća i broj izbačenih znakova.

Test primjeri:

```
char s[] = „Primjer tekst“  
n = proberi(s, 'k');
```

Rezultat: u **s** sada imamo „**Primjer tekst**“, a varijabla **n** == 1

Rješenje: **proberi.c**

1.4. Okreni niz znakova

Napisati funkciju koja okrene sadržaj niza znakova.

Test primjeri:

```
char s[] = „Tekst 1!“  
okreni(s);
```

Rezultat: u **s** sada imamo „**!1 tskeT**“

Rješenje: **okreni.c**

1.5. Pomicanje polja

Napisati program koji učitava 11 cijelih brojeva te nakon toga ispisuje 10 unesenih brojeva pomaknutih za broj mjesta koji određuje 11. broj. Problem riješiti korištenjem polja i funkcije. Test primjeri:

Ulaz: 1 2 3 4 5 6 7 8 9 10 -2
Izlaz: 3 4 5 6 7 8 9 10 1 2
Ulaz: 1 2 3 4 5 6 7 8 9 10 2
Izlaz: 9 10 1 2 3 4 5 6 7 8

Rješenje: pomakf.c

1.6. Pomicanje stupaca matrice

Napisati program koji pozivom funkcije pomiče zadanu matricu NxM (određuje korisnik) za jedno mjesto po stupcima tako da zadnji stupac postane prvi. Matrica može imati najviše 10 stupaca i 5 redaka. Matrica može sadržavati cijele brojeve ili znakove.

Test primjer:

Ulaz:	Izlaz
0 1 2 3 4 5	Pomaknuta matrica: 2 0 1 5 3 4

1.7. Sumiranje matrice

Napisati program koji pozivom funkcije sumira zadanu matricu NxM (određuje korisnik) po stupcima, po retcima i ukupno. Matrica može imati najviše 10 stupaca i 5 redaka. Matrica može sadržavati cijele brojeve ili znakove.

Test primjer:

Ulaz:	Izlaz
0 1 2 3 4 5	30 3 5 7 3 0 1 2 12 3 4 5

1.8. Okretanje kvadratne matrice

Napisati program koji pozivom funkcije okreće zadanu kvadratnu matricu NxN u smjeru kazaljke na satu:

Matricu je potrebno učitati i okrenuti je 1, 2 ili 3 puta

Potrebno je ispisati originalnu i tri stvorene matrice

Problem ograničiti na maksimalnu dimenziju 10x10

Matrica može sadržavati cijele brojeve ili znakove

Test primjer:

Ulaz:	Izlaz
0 1 2 7 8 3 6 5 4 2	1. okretanje matrice: 6 7 0 5 8 1 4 3 2 2. okretanje matrice: 4 5 6 3 8 7 2 1 0

Rješenje: rotmatf.c

1.9. NAJBROJ

Potrebno je napisati program koji će od unesenih realnih brojeva s tipkovnice odrediti najveći i najmanji korištenjem funkcijskih poziva

Rješenje: **najbroj.c**

1.10. INIT-P

Potrebno je napisati program koji će inicijalizirati jednodimenzionalno cjelobrojno polje preko funkcijskog poziva na više načina.

Rješenje: **init-p.c**

1.11. PISMO GLAVA

Potrebno je napisati dva programa koji će simulirati bacanje novčića.

Prvi program treba simulirati odabrani broj bacanja (npr. 1000) i odabir te na kraju iskazati broj pogodaka.

Drugi program treba po volji puta dati korisniku da odabere pismo ili glavu te obavještavati korisnika je li pogodio slučajno generiranu vrijednost. Na kraju treba ispisati broj pogodaka.

Oba programa na kraju trebaju ispisati i postotak uspješnosti.

Rješenje: **pismoglava.c**

1.12. BROJIZ

Potrebno je napisati program koji će prebrojati znakove u formatiranoj datoteci i to ispisati kao rezultat.

Rješenje: **brojiz.c**

1.13. TRAZI BROJ

Potrebno je napisati program koji će tražiti cijeli broj (od 1 do 999 uzlazno) u formatiranoj slijednoj datoteci. Program otvara datoteku čita brojeve, korisnik unosi traženi broj, i program traži broj. Rezultat traženja je da broja nema u datoteci ili redni broj na kome se broj nalazi.

Program završava kada korisnik unese -1.

Datoteka brojevi.txt se može formirati text editorom ili automatski.

Napraviti i varijaciju koja ne čita cjelokupni sadržaj već uvijek pretražuje datoteku.

Rješenje: **trazi.c**

1.14. TAB U PRAZNU

Potrebno je napisati program koji će za ulaznu text datoteku zamijeniti sve tab znakove u praznine. Zamjenu napraviti tako da se koristi privremena datoteka i konačni rezultat zamjene spremi u početnu datoteku.

Test datoteka se može formirati text editorom.

Rješenje: **tab2space.c**

1.15. IMENIK

Potrebno je napisati program koji će kreirati formatiranu datoteku za spremanje imena i ocjene (float).

Prvi program treba kreirati formatiranu datoteku imenao.txt tako da iz formatirane datoteke s listom imena (imena.txt) uzme ime i od korisnika zatraži ocjenu. Na kraju treba ispisati broj obrađenih imena.

Drugi program treba omogućiti pregledavanje ukupnog sadržaja datoteke s imenima i ocjenom.

Rješenje: **imenik0.c i imenik1.c**

1.16. EVIDENCIJA

Potrebno je napisati program koji će kreirati neformatiranu datoteku za spremanje i pretraživanje broja (int), prezimena (15 znakova), imena (10 znakova), prisutnosti (int) i prosjeka bodova (float).

Prvi program treba kreirati neformatiranu direktnu datoteku evidencija.dat te ju ispuniti sa 100 praznih zapisa.

Drugi program treba omogućiti pregledavanje, dodavanje te izmjenu svih podataka. Uz to treba na ekranu za postojeći ili novo uneseni zapis izračunati iz prisutnosti i prosjeka bodova

Rješenje: **evidencija0.c i evidencija1.c**

1.17. MALI TELEFONSKI IMENIK

Potrebno je napisati program koji će kreirati formatiranu datoteku za spremanje i pretraživanje imena (20 znakova), telefonskog broja (13 znakova) i dodatnog broja (cijeli broj od maksimalno 7 znamenki). Program treba omogućiti dodavanje, traženje i listanje svih podataka.

Rješenje: **adresar.c**