

Knjžara je dobila novi prostor i sve knjige treba prepakovati u nove police. Svaka polica je određena za po jedan žanr. U svaku policu staje najviše N knjiga. Za svaku knjigu na policu se stavlja najviše 10 primeraka, a ostali primerci, ako ih ima, se stavljaju u magacin. Kada se polica sa određenim žanrom popuni, sve naredne knjige tog žanra se stavljaju u magacin. Za svaku knjigu postoji informacija koliko primeraka knjige postoji u knjižari i koliko primeraka je prodato od početka godine. Nakon što se rasporede sve knjige, bira se 10 najprodavanijih knjiga (tj. 10 knjiga sa najvećim brojem prodatih primeraka od početka godine) i one se sa "žanr-police" prebacuju na "top10-policu", a prazna mesta na "žanr-polici" se dopune novim knjigama iz magacina. Glavni menadžer knjižare jednom nedeljno, na osnovu informacija o prodanim knjigama u protekloj nedelji, popunjava "žanr-police" novim primercima i/ili novim knjigama iz magacina. Ukoliko je potrebno tom prilikom se menja i sadržaj "top10-police". Knjige koje se sklanjaju sa "top10-police", ukoliko je odgovarajuća "žanr-polica" popunjena, se smešaju u magacin.

Napisati program koji učitava broj knjiga koje mogu stati na jednu policu N , zatim se učitava broj različitih knjiga koje se nalaze u knjižari i za svaku knjigu naslov, autora, žanr, broj primeraka prodatih od početka godine, broj primeraka koje knjižara ima. Program treba da izvrši raspoređivanje knjiga po policama (prvo po "žanr-policama, a potom "top10-polici"). Za posmatranu nedelju unosi se broj različitih knjiga koje su prodane, a zatim za svaku knjigu naslov i autor knjige, broj primeraka koji je prodat tokom te nedelje. Program treba da predvidi da će glavni menadžer posao organizovanja knjiga raditi K uzastopnih nedelja, pri čemu se K učitava. Na kraju ispisati koliko je ukupno primeraka različitih knjiga ostalo u magacinu.

(3 poena)

Za rešavanje problema napisati sledeće funkcije:

a) Napisati funkciju *Zanr* koja pronalazi zadati žanr u spisaku žanrova ili dodaje novi žanr, ukoliko je potrebno.

(3 poena)

b) Napisati funkciju *DodajKnjigu* koja učitano knjigu smešta na odgovarajuću "žanr-policu" i/ili u magacin.

(2 + 4 poena)

c) Napisati funkciju *UcitajKnjige* koja iz datoteke učitava broj knjiga i za svaku knjigu naslov, autor, žanr, broj primeraka prodatih od početka godine, broj preostalih primeraka.

(3 poena)

d) Napisati funkciju *Top10* koja prebacuje najprodavanije knjige na "top10-policu" i vrši popunu upražnjenih mesta na ostalim policama.

(4 poena)

e) Napisati funkciju *Prodato* koja učitava koliko je različitih knjiga prodato u poslednjih nedelju dana i za svaku knjigu prodatu knjigu naslov, autora i broj prodatih primeraka i menja brojno stanje za svaku knjigu. Knjiga čije je brojno stanje 0 se sklanja iz spiska knjiga.

(3 poena)

f) Napisati funkciju *Top10Repack* koja će organizovati sadržaj "top10-police" nakon informacija o prodaju iz prethodne nedelje.

(4 poena)

g) Napisati funkciju *Magacin* koja će odrediti ukupan broj primeraka knjiga koje se nalaze u magacinu.

(4 poena)

Napomene.

Zadatak rešiti bez korišćenja globalnih promenljivih.

U knjižari je kombinacija naslov knjige i autor jedinstvena. Nikada neće biti prodato više knjiga nego što ih ima na stanju.

Strukture podataka i algoritmi 1
Test – max 20 poena

Oktobar, 2017

1. Definirati strukturu za čuvanje podataka o knjigama iz zadatka o knjižari, vodici računa o tome da postoji i informacija gde je knjiga smeštena.
2. Definirati funkciju **Zanr** iz zadatka o knjižari.
3. Napisati test primer kojim bi se testirao program iz zadatka o knjižari.
4. Šta je rezultat sledećeg koda i zašto?

```
#include <stdio.h>
int main() {
    float a = 12.5;
    printf("%d\n", a);
    printf("%d\n", *(int *)&a);
    return 0;
}
```

5. Na 3 primera objasniti šta i kako radi sledeća funkcija

```
int fun(unsigned int x) {
    int count=0;
    while(x) {
        count++;
        x = x&(x-1);
    }
    return count;
}
```

6. Odrediti vrednost promenljive x posle izvršenja sledećih nizova naredbi

```
int x=1;
int y=0;
int z;
z = ((-x) ? 2 : -2) + 2;
x = z+++y;
```

7. Navesti oznake ispred "brzi" algoritma za sortiranje niza

- | | | |
|--------------|------------------|-----------|
| a. Selection | d. Stack | g. Merge |
| b. Quick | e. Odd-even sort | h. Binary |
| c. Shell | f. Insertion | |

8. Dopisati delove koda koji nedostaju tako da se u funkciji **f** učitaju celobrojne vrednosti koje će biti prikazane naredbom **printf** u **main**-u.

```
#include <stdio.h>
main() {
    int a,*b,**c;

    a=f(b,c);

    printf("%d %d %d\n",a,*b,**c);
}
```

9. Pomoću direktive **define**, definisati funkciju koja određuje treći stepen argumenta.

10. Kada i kako se koriste ključne reči **register** i **extern**.

11. Šta je opisano sledećim deklaracijama:

- a. `char (*f)[5];`
- b. `int ((*f())[ ] )();`
- c. `typedef int (*PFI)(char *, char *);`