

Strukture podataka i algoritmi 1

(zadatak - max 30 poena)

Avgust, 2018

Rsetoran u svom magacinu ima sastojke potrebne za pripremu jela i pića iz menija. Za sastojke se vodi precizna evidencija o nazivu sastojka (niz karaktera) i količini u gramima (realan broj). U magacinu ne postoje dva sastojka sa istim nazivom. Za ponude iz menija postoje recepti koji sadrže naziv ponude (niz karaktera), broj sastojaka od kojih se pripremaju (ceo broj), naziv svakog sastojka i njegova količina, u gramima, potrebna za pripremu (realan broj) i cena (realan broj). U meniju ne postoje dve ponude sa istim imenom i svaka ponuda se priprema po receptu, bez odstupanja. Ukoliko se u meniju nalazi ponuda za čiju pripremu ne postoji sastojak u magacinu, taj sastojak se dodaju na spisak sastojaka u magacinu sa količinom 0. Porudžbine se primaju navođenjem naziva ponude iz menija i brojem komada, pri čemu nije moguće naručiti ništa van menija. Ukoliko za pripremu neke ponude u magacinu nije bilo nekog sastojka u datoteku *Nabaviti.txt* se upisuje naziv sastojka koji treba poručiti. Za svako pripremljenu ponudu u magacinu se ažurira količina odgovarajućih sastojaka. Na kraju dana ispisuje se ponuda iz menija koje je najviše puta bila pripremljena, koliko je ukupno naplaćeno za sve narudžbine, pri čemu se, naravno, ne računaju one koje nisu mogle biti pripremljene zbog nedostatka sastojaka, spisak sastojaka koje treba nabaviti za magacin i učitava se sa ulaza ponuda iz menija čiji se svi podaci brišu.

(3 poena, obavezno)

Za rešavanje problema napisati sledeće funkcije:

- a) Napisati funkciju **UcitajMagacin** koja iz u datoteke učitava i formira niz sastojaka iz magacina i njihovih količina.
(3 poena, obavezno!)
- b) Napisati funkciju **NadjiSastojak** koje vraća pokazivač na odgovarajući sastojak iz magacina. Ukoliko sastojak ne postoji, dodati ga.
(3 poena)
- c) Napisati funkciju **UcitajMeni** koja iz datoteke učitava podatke o ponudama iz menija i formira niz ponuda sa pokazivačima na odgovarajuće sastojke.
(4 poena)
- d) Napisati funkciju **Pripremi** koja za ponudu iz menija proverava da li može biti pripremljeno i ukoliko može ažurira podatke u magacinu, a ukoliko ne može u datoteku *Nabaviti.txt* upisuje namirnice koje nedostaju.
(4 poena)
- e) Napisati funkciju **Porudzbine** koja sa standardnog ulaza učitava porudžbine sve dok ne učitava reč KRAJ i realizuje porudžbine vršeći sva potrebna ažuriranja.
(4 poena)
- f) Napisati funkciju **NajvisePripremljeno** koja će pronaći ponudu iz menija koja je najviše puta pripremljena.
(2 poena)
- g) Napisati funkciju **UrediNabavku** koja će sređuje datoteku *Nabaviti.txt* tako da se u njoj svaki sastojak nalazi po jedanput i ispisuje sadržaj datoteke.
(4 poena)
- h) Napisati funkciju **ObrisiPonudu** koja će iz niza ponuda obrisati sve podatke o datoj ponudi..
(3 poena)

Zadatak rešiti bez korišćenja globalnih promenljivih i bez unapred definisanih dužina korišćenih nizova.

Strukture podataka i algoritmi 1

Test – max 20 poena

Augst, 2018

1. Definirati strukture za čuvanje podataka o sastojcima i ponudama iz menija iz zadatka o restoranu, vodeći računa o predviđenim pokazivačima.
2. **(Obavezno!)** Definirati funkciju **UcitajMagacin** iz zadatak o restoranu.
3. Definirati funkciju **NajvisePripremljeno** iz zadatka o restoranu.

4. Šta je rezultat sledećeg koda

```
#include <stdio.h>
main()
{
    int x;
    scanf("%d",&x);
    if(x < 20)
        if(5)
            x &= 012 + 1;
    else
        x=-10;
    printf("%d\n",x);
}
```

Ako se kao vrednost promenljive x unese:

- a. 12 b. 5 c. 30

5. Šta je rezultat sledećeg koda?

```
a. #include <stdio.h>
void f1(int *a)
{
    printf("%ld",sizeof(a));
}
main()
{
    int x[10];
    printf("%ld",sizeof(x));
    f1(x);
}

b. #include <stdio.h>
int main()
{
    int x=1;
    int y=0;
    int z;
    z = ((-x) ? 3 : -3) + 1;
    x = z+++y;
    printf("%d %d %d",x,y,z);
}
```

6. Neka je data sledeća definicija

```
int **a;
```

Napisati deo koda koji će učitati i formirati kvadratnu matricu od N vrsta (N se učitava na početku) i sve učitane vrednosti pamti u promeljivoj a.

7. Data je sledeća definicija

```
enum vrsta {KVADRAT, PRAVOUGAONIK};
struct {
    vrsta tip;
    union {
        float a;
        struct {
            float a;
            float b;
        } stranice;
    } fig;
} niz[2];
```

Dodeliti vrednosti svakom elementu niza niz, tako da elementi sadrži informacije o različitim figurama.

8. Dopisati sve delove koda koji nedostaju, uključujući i definiciju funkcije `f`, 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. Na primeru objasniti šta i kako radi sledeća funkcija

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

10. Definirati 4 promenljive i inicijalizovati ih sledećim vrednostima, pri čemu se svi dati brojevi pamte kao neki od numeričkih tipova:

- 57.9
- Andersenove bajke
- 18, 67, 48, 4783695698, 15478
- Srpski jezik, 5, Matematika, 4, Fizičko, 5

11. Definirati enumerisani tip podataka gde će konstante JEDAN, CETIRI, STO, HILJADU imati numeričke vrednosti koje odgovaraju njihovom imenu.

12. Šta je opisano sledećim deklaracijama:

- `char* (*f)();`
- `void (*f[10]) (int, int);`
- `typedef int (*PFI)(char *, char *);`