|  |  |
| --- | --- |
| Univerzitet u Sarajevu  Elektrotehnički fakultet Sarajevo  Osnove računarstva | Rok za slanje zadaće:  *20. 1. 2016. (srijeda)* |

Zadaća 5

*Zadaća nosi 2 boda, a sastoji se od jednog zadatka. Zadatak će biti bodovan proporcionalno broju uspješnih testova. Oblasti zadaće su: strukture i datoteke (P18 i P19), ali se pretpostavlja i poznavanje prethodnog gradiva.*

*U ovoj zadaći je potrebno pridržavati se i pravila za pisanje čitljivog koda (P13), u suprotnom studenti mogu gubiti bodove.*

*Važna napomena: Ovaj dokument će možda biti mijenjan radi popravljanja grešaka. Koristite najnoviju verziju! Na dnu dokumenta nalaziće se dnevnik izmjena.*

**1. (2 boda)** Potrebno je napraviti program koji nekoj firmi olakšava upravljanje skladištima.

Firma posjeduje određeni broj skladišta. Korisnik može dodavati, mijenjati i brisati skladišta na nekom spisku skladišta. U firmu pristiže roba u vidu paketa. Svaki paket sadrži određenu količinu neke robe, pri čemu se u jednom paketu može nalaziti samo jedna vrsta robe. Potrebno je unijeti paket i dodijeliti ga u odgovarajuće skladište. Također je moguć izlaz robe (brisanje iz skladišta). Svi podaci se trebaju čuvati u datotekama radi mogućnosti nastavka rada u slučaju nestanka struje, zatvaranja programa i slično.

**DETALJI:**

Program treba čuvati podatke u dva *globalna niza struktura* koji su definisani na sljedeći način:

struct Paket {

char naziv\_robe[100];

int kolicina;

float cijena;

} paketi[1000];

struct Skladiste {

char naziv[100];

int kapacitet;

struct Paket\* sadrzaj[100];

int br\_paketa;

} skladista[100];

Pored ovoga trebaju postojati dvije globalne promjenljive br\_paketa i br\_skladista u kojima se nalazi broj članova ova dva niza.

Objašnjenja polja strukture Paket su sljedeća:

* **naziv\_robe** - npr. "brasno", "krompir", "cement",
* **kolicina** - količina robe u kilogramima,
* **cijena** - cijena jednog kilograma date robe u KM. Npr. ako je kolicina=300, cijena=0,6, ukupna cijena robe u paketu je 180 KM.

Objašnjenja polja strukture Skladiste su sljedeća:

* **naziv** - ime koje je dato skladištu,
* **kapacitet** - maksimalna količina robe u kilogramima koja se može pohraniti u skladište,
* **sadrzaj** - niz pokazivača na pakete koji se nalaze u skladištu,
* **br\_paketa** - broj članova niza sadrzaj.

Pored onoga što se vidi iz definicije struktura, podaci moraju uvijek pratiti sljedeća **ograničenja**:

* ne smiju postojati dva skladišta sa istim nazivom;
* jedan paket se ne može nalaziti u više skladišta istovremeno niti se može dvaput staviti u isto skladište;
* ukupna količina robe u svim paketima u nekom skladištu ne može biti veća od kapaciteta tog skladišta;
* količina, cijena, kapacitet ne mogu biti negativni brojevi niti nula.

Program mora sadržavati sljedeće funkcije:

* **int ucitaj\_datoteke()** - učitava podatke iz datoteka pod imenima "skladista.dat" i "paketi.dat" u globalne nizove. Datoteke se moraju nalaziti u tekućem direktoriju. Struktura i format datoteka je prepuštena vama. Funkcija vraća logičku istinu ako učitavanje uspije, a neistinu ako ne uspije.
* **int zapisi\_datoteke()** - zapisuje podatke iz globalnih nizova u datoteke. Povratna vrijednost kao iznad.
* **int dodaj\_skladiste(const char\* naziv, int kapacitet)** - dodaje novo skladište sa datim nazivom i kapacitetom u globalni niz skladista. Vraća indeks člana u nizu ili -1 ako skladište nije uspješno dodato jer nisu zadovoljena ograničenja.
* **int promijeni\_skladiste(int indeks, const char\* naziv, int kapacitet)** - mijenja podatke o skladištu čiji je indeks indeks sa podacima koji su dati kao parametri. Vraća logičku istinu ako je promjena uspjela ili logičku neistinu ako skladište sa datim indeksom ne postoji ili ako traženom promjenom više ne bi bila zadovoljena ograničenja.
* **int obrisi\_skladiste(int indeks)** - briše iz globalnog niza skladište čiji je indeks indeks. Pri tome ostala skladišta u nizu trebaju zadržati redoslijed. Vraća logičku istinu ako je brisanje uspjelo ili logičku neistinu ako skladište sa datim indeksom ne postoji.
* **int dodaj\_paket(const char\* naziv\_robe, int kolicina, float cijena)** - dodaje novi paket sa navedenim podacima u niz paketi, a vraća indeks člana u nizu ili -1 ako paket nije uspješno dodat jer nisu zadovoljena ograničenja.
* **int stavi\_paket\_u\_skladiste(int paket, int skladiste)** - stavlja paket u skladište, pri čemu su parametri indeksi u nizovima paketi i skladista. Funkcija mora korektno ažurirati sve podatke i paziti da su ispunjena **ograničenja** koja smo naveli ranije. Vraća logičku istinu ako je paket uspješno dodat, a neistinu ako nije (nisu zadovoljena ograničenja ili su parametri neispravni).
* **int izbaci\_paket\_iz\_skladista(int paket, int skladiste)** - vadi paket iz skladišta, pri čemu su parametri indeksi u nizovima paketi i skladista. Funkcija mora korektno ažurirati sve podatke i paziti da su ispunjena **ograničenja** koja smo naveli ranije. Pri tome ostali paketi u nizu trebaju zadržati redoslijed. Vraća logičku istinu ako je paket uspješno izbačen, a neistinu ako nije (parametri su neispravni ili se paket ne nalazi u skladištu).
* **int obrisi\_paket(int paket)** - briše paket sa navedenim podacima iz niza paketi, pri čemu se paket i uklanja iz skladišta ako se eventualno nalazi u nekom skladištu. Pri tome ostali paketi u nizu trebaju zadržati redoslijed. Pošto bi brisanje iz niza paketa promijenilo adrese pokazivača, "brisanje" treba izvesti tako što će se količina postaviti na nulu. Prilikom dodavanja sljedećeg paketa on se može pisati preko paketa čija je količina nula. Funkcija vraća logičku istinu ako je brisanje uspjelo a neistinu ako paket sa datim indeksom ne postoji.
* **int ulaz\_robe(const char\* naziv\_robe, int kolicina, float cijena)** - ova funkcija treba kreirati novi paket sa datim podacima, a zatim ga dodati u prvo skladište (gledajući po indeksima u nizu skladišta) u kojem ima dovoljno mjesta za ovu robu. Ako nigdje nema dovoljno mjesta ili nije ispunjeno neko ograničenje, funkcija treba vratiti logičku neistinu, a u suprotnom logičku istinu.
* **float izlaz\_robe(const char\* naziv\_robe, int kolicina)** - funkcija treba pronaći u kojem skladištu se nalazi paket sa datim podacima. Ako se u nekom skladištu nalazi paket sa datim nazivom robe i količinom, treba obrisati paket. Ako se nalazi paket sa datim nazivom ali većom količinom, treba ažurirati paket tako da mu se smanji količina robe za navedeni iznos. U oba ova slučaja funkcija treba vratiti ukupnu cijenu te robe za tu količinu. Ako niti u jednom skladištu ne postoji paket sa datim nazivom i istom ili većom količinom robe, funkcija treba vratiti broj -1.
* **void statistika()** - ova funkcija treba na ekranu ispisati statističke podatke o skladištima u sljedećem formatu:

Evidentirana su 3 skladista sa ukupnim kapacitetom 500kg.

Od toga je popunjeno 120kg (24.00%).

Ukupna vrijednost robe u skladištima je 384.20 KM.

Možete u programu po želji imati i neke druge funkcije osim navedenih. Svaka funkcija može pozivati druge funkcije (ali ne mora).

Pored nabrojanih funkcija, program mora sadržavati i main funkciju koja korisniku omogućuje da koristi nabrojane funkcije, a podaci se trebaju čitati iz datoteka i pisati u datoteke. Program treba raditi ispravno ako datoteke ne postoje (kreirati prazne datoteke), a ako dođe do neke druge vrste greške treba prekinuti program sa izlaznom vrijednošću 1. Prilikom pokretanja program treba najprije ponuditi korisniku sljedeći meni:

1) Skladista

2) Ulaz robe  
3) Izlaz robe  
4) Statistika  
0) Izlaz  
Unesite vas izbor:

Korisnik unosi broj 0-4, u slučaju da se unese broj izvan tog opsega ponavlja se prikaz menija. Broj 0 završava program. Broj 1 prikazuje podmeni "Skladišta" koji izgleda ovako:

1) Spisak skladista

2) Podaci o skladistu  
3) Novo skladiste

4) Promijeni skladiste

5) Obrisi skladiste  
0) Povratak na glavni meni

Unesite vas izbor:

Unos je kao i ranije. Nakon završetka svake opcije ponovo se prikazuje podmeni "Skladišta" sve dok korisnik ne izabere povratak na glavni meni. Opcija "Spisak skladišta" ispisuje nazive skladišta, svaki u zasebnom redu. Opcija "Podaci o skladištu" izgleda ovako (dio sa žutom pozadinom unosi korisnik):

Unesite naziv skladista: Novo Sarajevo

Kapacitet skladista: 150kg  
Popunjeno: 150kg (100.00%)

Paketi:

- brasno (20kg, 0.60 KM/kg)

- brasno (30kg, 0.50 KM/kg)

- krompir (100kg, 0.30 KM/kg)

Može se vidjeti da naziv skladišta može sadržavati razmake. U slučaju da se unese nepostojeće skladište, treba ispisati poruku "Nepostojece skladiste".  Podaci o paketima trebaju biti ispisani onim redom kojim su dodavani u skladište.

Opcija "Novo skladište" izgleda ovako:

Unesite naziv skladista: Novo Sarajevo

Unesite kapacitet skladista: 150

Opcija "Promijeni skladište" treba izgledati ovako:

Unesite naziv skladista koje mijenjate: Novo Sarajevo  
Unesite novi naziv: Stari Grad

Unesite novi kapacitet skladista: 180

Opcija "Obriši skladište" treba izgledati ovako:

Unesite naziv skladista koje brisete: Novo Sarajevo

Skladiste uspjesno obrisano.

U slučaju da su podaci u bilo kojem pogledu neispravni, ne zadovoljavaju ograničenja itd. treba ispisati poruku "Neispravni podaci" ili "Nepostojece skladiste".

Opcija sa glavnog menija "Ulaz robe" poziva funkciju ulaz\_robe, a treba izgledati ovako:

Unesite naziv robe: vjestacko gnojivo  
Unesite kolicinu robe: 200

Unesite cijenu robe: 1.1

Može se uočiti da naziv robe može sadržavati razmake. U slučaju da ulaz robe ne uspije treba ispisati poruku "Ulaz robe nije uspio" i vratiti se na glavni meni. Slično radi i opcija "Izlaz robe". Konačno opcija "Statistika" treba samo pozvati funkciju statistika koju smo opisali ranije.

Sa ranije napravljenim funkcijama bi se dalo dodati još opcija u menije, ali u ovoj zadaći to nećemo raditi da ne bi bila predugačka :)

*Izmjene:*

* *8.1.2016 18:35 - u meni opciji "Podaci o skladištu" precizirano da cijena robe ide na dvije decimale (novčani iznosi bi u pravilu uvijek trebali imati dvije decimale jer je to iznos feninga).*
* *8.1.2016 20:00 - ispravka u definiciji funkcije obrisi\_paket kako bi se izbjeglo da se svi pokazivači u svim skladištima moraju ažurirati (što bi bilo izvodivo ali puno posla).*