

POČÍTAČOVÉ CVIČENÍ 7

Cílem cvičení je procvičit si práci se strukturami, uniemi a dynamickými proměnnými v jazyku C.

1. Doplňte zdrojový kód programu k příkladu **C07_1.cpp** o funkce pro správu databáze zelinářství. Program využívá několik typů struktur, které jsou již předdefinovány. V první řadě je to struktura typu **t_frtvgtbl**, která zapouzdřuje položky pro daný druh ovoce či zeleniny na skladě. Význam je zřejmý ze zdrojového kódu. Zajímavá je položka struktury **qtype**, kterou je definováno, zda je daný druh ovoce nebo zeleniny počítán a prodáván na kilogramy nebo kusy. Z tohoto důvodu je pro položku obsahující množství vytvořena unie typu **t_quantity**. Celý program je postaven na dynamickém vkládání jednotlivých druhů ovoce a zeleniny do skladu, k čemuž slouží struktura **t_record** obsahující ukazatele na jednotlivé vložené záznamy **t_frtvgtbl** a položku **status** definující, zda byl vytvořen a naplněn daný záznam **t_frtvgtbl**. Následně je alokováno globální pole struktur **t_record** pro maximálně 30 druhů ovoce a zeleniny. Pořadí v tomto poli odpovídá i kódu daného ovoce či zeleniny. Nové záznamy lze vkládat na libovolné prázdné místo podle požadavků (neobsazeného kódu, který chcete přidělit danému druhu ovoce a zeleniny – tak jak by to bylo například na váhách v supermarketu). V programu je rovněž definována struktura **t_bill** pro nákup obsahující kód a zakoupené množství daného zboží ze skladu. Při vyúčtování nákupu pak lze podle kódu nalézt požadované hodnoty jako je druh zboží, cena za kg nebo kus pro výpočet celkové útraty a pochopitelně také pro snížení stavu daného ovoce či zeleniny ve skladu. Zdrojový kód programu obsahuje deklarace všech struktur a globálních proměnných, hotový hlavní program s uživatelským rozhraním a vybrané funkce: tisk stavu skladu **print_store()** a kompletace účtu při nákupu **bill_sum()**. Mimo tyto funkce je do programu přidána funkce **init_recs()**, která částečně naplní sklad z předdefinovaných položek na začátku programu (aby programátor při ladění nemusel neustále vkládat nějaké zásoby). Zbývající funkce je třeba dokončit a to funkci, která přidá nový záznam ovoce nebo zeleniny (se všemi položkami) pojmenovaná **add_newitem()**, dále funkci **add_item()**, která navýší skladové zásoby daného druhu ovoce nebo zeleniny, a funkci **sell_item()**, která přidá položku nákupu do košíku (tedy do pole **bill[]**). Poslední pomocná funkce **print_freeitems()** zobrazí seznam volných kódů (míst v poli **rec[]**) tak, aby při zavolání funkce **add_newitem()** obsluha věděla, který kód je ještě volný. Všechny hodnoty od obsluhy se zadávají přímo v příslušných funkcích.

Doporučený postup řešení:

- a) Sestavit funkci **add_item()** a otestovat, že správně přidává již zavedené zboží ve skladu.
- b) Sestavit funkci **sell_item()** a otestovat funkčnost nakupování, tzn. že nakoupené zboží se odečítá ze skladu a všechny nakoupené položky se objeví na závěrečném účtu vč. správně odebrané hmotnosti či kusů a ceny.
- c) Sestavit funkci **print_freeitems()** a otestovat, že skutečně vypíše seznam volných kódů (míst v poli **rec[]**). V první fázi můžete funkci volat v hlavním programu místo vložení nového druhu. Po splnění bodu d) tuto funkci volejte přímo v **add_newitem()**. Pozn. kódy jsou vždy zadávány od 1 (ani na váze v Bille není kód 0), indexy v poli od 0.
- d) Sestavit funkci **add_newitem()** s dynamickým vložením nového záznamu pro nový druh ovoce či zeleniny ve skladu. Tiskem obsahu skladu zjistíte, zda jsou všechny položky záznamu správně naplněny.

2. Neveřejný příklad zadaný cvičícím.

HODNOCENÍ: Za splnění prvního příkladu 1 bod, za splnění druhého příkladu 1 bod.

```
d:\Dokumenty\Visual Studio 2008\Projects\C07_1\Debug\C07_1.exe
Jahody <FRA>
code: 10 quantity: 45.23 kg price: 68.90 Kc/kg
Svestky <CZE>
code: 11 quantity: 15.17 kg price: 46.90 Kc/kg
Hrusky <CZE>
code: 12 quantity: 153.14 kg price: 39.90 Kc/kg

Insert command: 'q' = quit, 'p' = printing store, 'a' = adding goods, 'n' = new item, 's' = sell goods a
Insert code: 2
Current quantity of Mandarinka is 85.55 kg.
Insert additional quantity in kg: 14.69
New quantity of Mandarinka is 100.24 kg.
Insert command: 'q' = quit, 'p' = printing store, 'a' = adding goods, 'n' = new item, 's' = sell goods s
Insert code: 1
Pomeranc: insert quantity in kg: 3.33
Pomeranc: 3.33 kg a 29.90 Kc/kg suma: 99.57 Kc <'n' = next item, other = bill>n
Insert code: 6
Pomelo : insert quantity in pcs: 2
Pomelo : 2 pcs a 36.00 Kc/pcs suma: 72.00 Kc <'n' = next item, other = bill>n
Insert code: 11
Svestky : insert quantity in kg: 1.56
Svestky : 1.56 kg a 46.90 Kc/kg suma: 73.16 Kc <'n' = next item, other = bill>n
Insert code: 2
Mandarinka: insert quantity in kg: 2.26
Mandarinka: 2.26 kg a 33.90 Kc/kg suma: 76.61 Kc <'n' = next item, other = bill>n
Bill assembling:
-----
1. Pomeranc: 3.33 a 29.90 Kc/u sum: 99.57 Kc
2. Pomelo : 2.00 a 36.00 Kc/u sum: 72.00 Kc
3. Svestky : 1.56 a 46.90 Kc/u sum: 73.16 Kc
4. Mandarinka: 2.26 a 33.90 Kc/u sum: 76.61 Kc
-----
SUM incl. VAT <10 proc.> 321.34 Kc
VAT <10 proc.> 32.13 Kc
SUM excl. VAT 289.21 Kc
-----

Insert command: 'q' = quit, 'p' = printing store, 'a' = adding goods, 'n' = new item, 's' = sell goods p
Store list:
Pomeranc <ESP>
code: 1 quantity: 146.69 kg price: 29.90 Kc/kg
Mandarinka <MAR>
code: 2 quantity: 97.98 kg price: 33.90 Kc/kg
Grep cerveny <CUB>
code: 3 quantity: 45.85 kg price: 34.90 Kc/kg
Kokos <JAM>
code: 4 quantity: 163 pcs price: 45.00 Kc/pc
Grep bily <ESP>
code: 5 quantity: 63.85 kg price: 31.90 Kc/kg
Pomelo <CHN>
code: 6 quantity: 40 pcs price: 36.00 Kc/pc
Kivi <ITA>
code: 8 quantity: 956 pcs price: 4.00 Kc/pc
Jahody <FRA>
code: 10 quantity: 45.23 kg price: 68.90 Kc/kg
Svestky <CZE>
code: 11 quantity: 13.61 kg price: 46.90 Kc/kg
Hrusky <CZE>
code: 12 quantity: 153.14 kg price: 39.90 Kc/kg
```

Obr. 1. Příklad zobrazení výsledků příkladu v konzolovém okně.