and cash payments arrive in the mail), they are
iod (i.e., a month for some companies, a week for
tions (called "trans.dat" in Exercise 11.3) is
:" in Exercise 11.3), thus updating each account's
iese updating runs, the master file is rewritten as a
at the end of the next business period to begin the

ain problems that do not exist in single-file pro-
cur. A customer on the master file may not have
it business period, and therefore no record for this
irly, a customer who did make some purchases or
inity, and the company may not have had a chance

as a basis for writing a complete file-matching
ier on each file as the record key for matching pur-
records stored in increasing account number order.
me account number appear on both the master file
the transaction file to the current balance on the
d. (Assume that purchases are indicated by posi-
nents are indicated by negative amounts.) When
ut no corresponding transaction record, merely
Vhen there is a transaction record but no cor-
'**Unmatched transaction record for**
: from the transaction record).

, write a simple program to create some test data
the following sample account data:

Balance
348.17
27.19
0.00
-14.22

iles of test data created in Exercise 11.8. Use the

11.10 It is possible (actually common) to have several transaction records with the same record key. This occurs because a particular customer might make several purchases and cash payments during a business period. Rewrite your accounts receivable file-matching program of Exercise 11.7 to provide for the possibility of handling several transaction records with the same record key. Modify the test data of Exercise 11.8 to include the following additional transaction records:

Account number	Dollar amount
300	83.89
700	80.78
700	1.53

11.11 Write statements that accomplish each of the following. Assume that the structure

```
struct person {
   char lastName[ 15 ];
   char firstName[ 15 ];
   char age[ 4 ];
};
```

has been defined and that the file is already open for writing.

a) Initialize the file "**nameage.dat**" so that there are 100 records with **lastName** = **"unassigned"**, **firstname** = **""** and **age** = **"0"**.
b) Input **10** last names, first names and ages, and write them to the file.
c) Update a record; if there is no information in the record, tell the user **"No info"**.
d) Delete a record that has information by reinitializing that particular record.

11.12 You are the owner of a hardware store and need to keep an inventory that can tell you what tools you have, how many you have and the cost of each one. Write a program that initializes the file **"hardware.dat"** to 100 empty records, lets you input the data concerning each tool, enables you to list all your tools, lets you delete a record for a tool that you no longer have and lets you update *any* information in the file. The tool identification number should be the record number. Use the following information to start your file:

Record #	Tool name	Quantity	Cost
3	Electric sander	7	57.98
17	Hammer	76	11.99
24	Jig saw	21	11.00
39	Lawn mower	3	79.50
56	Power saw	18	99.99
68	Screwdriver	106	6.99
77	Sledge hammer	11	21.50