Data Structures (CIS 320)
M. Lowenthal

ASSIGNMENT #1- REVIEW

The "ABC Hardware Company" has hired you to write a program for it's Accounts Receivable department (A/R are accounts that owe money to the company because they have purchased items and have not yet paid for them)

There are two types of input data

1) A master file in ascending order by customer number (customer numbers are 4 digits long). The file also contains a
20 character customer name and a balance due.

2) A transaction file that contains records of each transaction. This file is also in ascending order by customer number.
There should be more than one transaction record per master record. Each related group of data in a file is called a
record. A record should be stored in a structure.

Each record starts with a character, "O" for order or "P" for payment. Each record also contains the four-digit customer
number, a four-digit transaction number, and up to three more values:

If the code is "O", the record contains the item ordered (20 characters), the quantity ordered (an integer) plus the cost of the item. (You must multiply to get the total cost)

If the code is "P", the record contains the amount of the payment.

You are to read in records one at a time from the two files and use the transaction file records to update the master file. Process all transaction records for each master record before going on to the next master record. If the transaction record contains an "O" in column 1, calculate the order amount and add it to the balance due. If the record contains a "P" in column 1, subtract the payment from the balance due. Keep a running total of the A/R Balance of ABC Company (That is the sum of the balances due for each customer.

Your program should....

Check for errors such as duplicate master records or records in the transaction file that do not appear in the master file.

After processing a master record and all it's transactions the program should prepare an invoice for each customer which lists the customer name, number, previous balance (balance before transactions were performed), all transactions, and the final balance due (balance after transactions are performed.

The output should look something like this...

```
CUSTOMER NAME      CUSTOMER NUMBER

                     PREVIOUS BALANCE    $ XXX.XX

TRANSACTION #      ITEM ORDERED          $ ORDER AMOUNT
TRANSACTION #      ITEM ORDERED          $ ORDER AMOUNT
TRANSACTION #      PAYMENT               $ PAYMENT AMOUNT
TRANSACTION #      ITEM ORDERED          $ ORDER AMOUNT

                     BALANCE DUE         $ XXX.XX
```

Don't forget, payments reduce the balance and orders increase it. You are to create your own data using at least 7 customers with an average of 5 transactions each.

Data Structures (CIS 220)
M. Lowenthal

ASSIGNMENT #2- MORE REVIEW

This program assumes that a company has warehouses in 5 cities: New York, Los Angeles, Miami, Houston, and Chicago. In each warehouse the company stocks three items. A record contains the amount of each item currently in each warehouse.

The record should contain an array of 5 warehouses with the name of each warehouse and an array of the 3 items with the amount that is currently in stock in the warehouse.

Initially the warehouses are empty. Read in a card containing the three prices of the three items. Then read in cards which may be any one of two types:

1. Shipment cards containing...

S city amt1 amt2 amt3

This data represents a shipment to a given city of amt1 of item1, amt2 of item2, and amt3 of item3.

2. Order cards containing...

O city amt1 amt2 amt3

This data represents an order in a given city of the respective amounts of the various items.

As each shipment card is read in, print it out, update the amounts in that warehouse and then print on the next line:

city amt1 amt2 amt3

As each order card is read in, print it out. If there is enough in stock at that warehouse of each item, update the amounts. If there is not enough, search the other warehouses for the one which has the most of that particular item, ship the amount needed from one warehouse to the other, printing out:

x of item y shipped from city1 to city2

and

city1 amt1 amt2 amt3

10% extra is to be charged for any item shipped from one city to another. If no single warehouse has enough extra to fill the order for a particular item, print out :

Order Unfilled

However any shipments of the previous items remain in effect.
In either case print out the adjusted...

city amt1 amt2 amt3

And if the order if filled print...

Price of Order: Price

Assignment 2.

Data

Price 1=$2.00 Price 2=$7.00 Price 3=$8.50

s	New York	23	14	1
s	Miami	25	25	25
s	Los Angeles	40	13	17
s	Houston	100	30	10
s	Chicago	42	23	19
s	New York	0	0	15
s	Miami	13	17	21
o	Los Angeles	15	10	15
o	New York	12	24	8
o	Houston	75	45	10
o	Chicago	20	15	15
o	New York	15	0	0
s	Los Angeles	10	20	10
s	Houston	0	30	40
o	New York	15	15	25
o	Chicago	75	30	40
s	New York	20	15	20
o	Houston	10	20	10

End of data

Data Structures (CIS 2[illegible])
M. Lowenthal

ASSIGNMENT #3- LINKED LISTS

You are to use Linked Lists to do this program.

The "XYZ Widget Store" receives shipments of widgets at various costs. The store's policy is to charge a 30% markup, and to sell widgets which were received earlier before widgets which were received later. This is called a First In First Out policy (FIFO).

Write a program using linked lists that reads in 3 types of input data and does the following:

A sales record which contains an "S" in column 1 and a quantity which represents the number of widgets sold.

A receipt record which contains an "R" in column 1 and a quantity and a price which represents the receipt of a quantity of widgets at the stated cost per widget.

A promotion card which contains a "P" in column 1 and a number such as 25 which would represent a 25% discount to the next 2 buying customers (the next 2 sales cards)

The program should...

Print a message after each receipt record is read in with the price of the widgets received.

Print a message after each promotion card is read in with the amount of discount the next two customers will be receiving.

After a sales record is read in, print a message stating the number sold and the price of each widget and total price of the order. For example if 200 widgets were sold and there were 50 widgets at $1.00 and 100 at $2.00 and 50 at $3.00 print (recall the 30% markup and the FIFO policy)

```
200 Widgets sold
50  at  1.30 each     Sales: $65.00
100 at  2.60 each     Sales: $260.00
50  at  3.90 each     Sales: $195.00
                  Total Sales: $520.00
```

If there are insufficient number of widgets in stock to fill an order sell as many as are available and then print...

"remainder of xxx Widgets not available"

Do not forget the promotional discount

At the end of the data before exiting the program print out under a separate heading the widgets still left in stock and their original purchase price.

DATA / Assignment 3

R/S/P	# of widgets or discount %	Price
R	150	1.00
R	130	2.00
S	145	
R	50	2.50
S	75	
S	180	
R	50	4.00
R	30	5.00
R	40	5.50
P	30%	
S	50	
S	30	
R	50	6.00
R	265	10.00
S	60	
P	50%	
S	100	
S	70	
S	175	
R	40	14.00
R	75	15.00
S	110	
R	30	16.00
R	40	18.00

Data Structures (CIS 320)
M. Lowenthal

ASSIGNMENT #4 - TREES

Write a program which will process several sets of numbers.

For each set of numbers, you must...

1) Create a binary tree
2) Print the tree using "inorder", "preorder", and "postorder"
3) Call a subroutine "count", which returns the number of nodes in the tree
4) Call a subroutine "children" which prints the number of children each node has
5) Insert and delete several nodes according to the instructions given
6) Print the tree again using "inorder", "preorder", and "postorder"
7) Call subroutine "count" again, which returns the number of nodes in the tree
8) Call a subroutine "children" again, which prints the number of children each node has
9) Free the tree

To be done using dynamic storage and pointers.
To be done using static storage and arrays.

Data to be used: (-999 terminates the original data)

Set #1 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 -999
Insert 21 Delete 1 Insert 0 Delete 10
Delete 11 Delete 5 Delete 2 Insert 10

Set #2 3 1 5 -999
Delete 3 Delete 1

Set #3 -999
Delete 15 Insert 30 Insert 5 Insert 10 Insert 20
Delete 20 Delete 10 Delete 5 Delete 15 Delete 30

Set #4 2 -999
Delete 2

Set #5 11 25 75 12 37 60 90 8 15 32 45 50 67 97 95 -999
Delete 37 Delete 15 Insert 40 Insert 99

Set #6 50 40 60 30 70 20 80 10 90 -999

Set #7 30 40 20 10 50 -999

Note: Your insert/delete routines must be able to handle duplicate values and deleting non-existant values

Data Structures (CIS 220)
M. Lowenthal

ASSIGNMENT #5- MORE TREES

Write a complete program, using the node representation of binary trees (the nodes can be implemented using either an array or dynamic storage, your choice) to do the following...

The program will read in a set of data representing a family tree. It will implement this as a binary tree in Pascal, then it will answer a series of questions about the family tree.

Here is an outline of what the program will do:

First, it will do whatever initializations (if any) are necessary.

Then it will read in a set of data representing a family tree. The format of the data is described below. You should print the original data as it is read in. By hand, you should draw the original family tree in it's usual form.

The program will convert the data into a binary tree (this can be done as you are reading the data in). By hand you should draw the binary tree representing the original tree.

Next the program will answer a series of questions about the tree. Possible questions are: Given a person p in the tree...

1) Who is the father of p?
2) Who are all the sons of p?
3) Who are all the brothers of p?
4) Who is the oldest brother of p?
5) Who is the youngest brother of p?
6) Who is the oldest son of p?
7) Who is the youngest son of p?
8) Who are the uncles of p?
9) Who is the grandfather of p?

Note: P can be any node in the tree, including the root or a leaf. the first thing to do is locate p in the tree (you might want to store some information along the way down to p).

You should determine what the question is, then answer each question in a separate subprogram. (Of course, that subprogram may all other subprograms as well. Some of these may be shared.)

Note that the same question can be asked several times for a given tree, each time for a different node p. After you have answered a series of questions about this tree, you should start the process all over again for a different family tree. Start the output from each tree on a new page. Continue this for at least 5 family trees, including those given in class.

Your program should be written in a modular style. Be sure that each subprogram has a very good comment explaining what it does. Be sure to specify any global variables that you might use.

Assignment 5

DATA: The set of data for each tree will look like this (you decide how to separate sets of data for different trees):

a name (up to 10 characters) - This is the top node of the tree
an integer n (possibly 0), representing the number of sons this node has

Example:

Jones	3	(the root Jones has 3 sons)
Bob	2	
Dan	0	
Brian	1	(these are the 3 sons of jones)
Richard	0	
Jake	1	(those are the 2 sons of bob)
Michael	1	(the one son of brian)
Bill	0	(the one son of Jake)
Deville	0	(the one son of michael)

Note: You can use a different format, as long as you explain clearly what your format is

Optional: Answer questions on grandchildren, nephews, cousins, etc.

Data Structures (CIS 22~~0~~)
M. Lowenthal

ASSIGNMENT #6- Sorting

Write a complete program to collect statistics on a series of sorts.

The main program will continue the following until the end of the set of data...

1. Call a subprogram to read a group of numbers to be sorted
2. Call a subprogram to sort these values using three sorts
3. Call a subprogram to compare the three sorts

STEP 1: The read subprogram will first read a heading describing the group of numbers (eg the heading might be: 10 numbers in order, or 50 numbers in a random order or 100 numbers in reverse order). Then it will read in the group of numbers. The original group should be printed as it is read in.

STEP 2: The threesort subprogram will call three separate sorts (be sure to send each one the items in their original order):

a. One sort will be a stupid one - either linear or bubble
b. One sort will be quicksort
c. One sort will be any other good one of your choice (eg heapsort)

After each individual sort subprogram finishes, the sorted array should be printed. Also print the number of comparisons each sort needed and the of times elements were interchanged. (These values must be computed within the three individual sorting subprograms. Use either additional parameters or global variables, but be sure to include comments.)

STEP 3: The comparison subprogram will determine which sort used the most comparisons, which was in the middle, and which used the least. It will do the same thing for interchanges.

DATA: Remember that each group of data must start with a heading describing what it is testing. You are allowed to have repeats in the group, but there should not be too many repeated values. you should include at least the following 9 groups (the heading for a group can be what I use to describe it):

1. 10 numbers in almost sorted order (two or three numbers interchanged)
2. 10 numbers in random order
3. 10 numbers in reverse order
4. 50 numbers in almost sorted order (a few out of order)
5. 50 numbers in random order
6. 50 numbers in reverse order
7. 100 numbers in almost sorted order (a few out of order)
8. 100 numbers in random order
9. 100 numbers in reverse order

OPTIONAL: 1. Construct a sort which minimizes the number of interchanges (note: interchanges, not comparisons)
2. Construct a way to generate 50 (or 100) numbers, numbered 1-50 (or 1-100), without repeats, in random order.