Hands #6 - Solving Problems Involving Time and Money

Problem Types:

- **Convert AM/PM times to minutes**
- **Linear Searches**
- **Times as strings**
- **Finding percents**
- **Running totals**
- **Making change**
- **Amount of interest**
- **Rounding up to next unit**

On the following pages, there are 9 sample problems. The dat files and solutions to these problems are provided in a separate folder.

The solutions are based on Java 5.0. I have used meaningful variable names to make the code more readable. Students in contest situations would certainly use shorter variable names and possibly more anonymous variables.

Problem	**Key Concepts**
pr61 - Highs and Lows	time as a `String`; `compareTo`; linear search
pr62 - Mowing Fields	reading data on a single line; using data two at a time; divide and mod; `Math.round`
pr63 - Cell Phone 1	`split("[ :]")`; convert hours to minutes; military time; find elapsed time
pr64 - Duct Tape	cast; `Math.ceil`; `else-if` ladder; `printf` to format
pr65 - Compound Interest	convert percent to decimal; `printf` to format;
pr66 - Cash Returns	cast; mod; handling round off errors
pr67 - Fence Me In	cast; `Math.ceil`; `printf`;
pr68 - Cell Phone 2	`split("[ :]")`; convert hours to minutes; AM/PM; find elapsed time
pr69 - Grocery Bill	reading an undetermined amount of input; `charAt`; `switch`; running totals; percents; `printf` to format in columns and round;

pr61 - Highs and Lows

Problem: Write a program that will print the earliest time that the highest and lowest temperature occurred in a list of temperatures.

Input: The first line of the data set is an integer that represents the number of lines that follow. Each of the remaining lines contains a list of times and the temperature at that time. All times will be formatted as `hh:mm` with leading zeros as necessary to make two digit hours and minutes.

data file: `pr61`

Output: Output the lowest temperature and the time it occurred and the highest temperature and the time it occurred. If the lowest or highest temperature occurs more than once, output the earliest time that it occurred.

Assumptions: None.

Sample Input:

```
5
02:21 57 11:22 81 12:35 44 17:59 76
04:56 22 12:44 52 19:10 77 02:21 72 05:48 80 04:22 77
05:34 44 03:45 88 19:23 44 04:45 89 08:35 44
15:34 44 03:45 88 09:23 44 04:45 89 08:35 44
04:56 22 12:44 22 19:10 22 02:21 22
```

Output Input:

```
LOW:  44 OCCURRED AT 12:35
HIGH: 81 OCCURRED AT 11:22

LOW:  22 OCCURRED AT 04:56
HIGH: 80 OCCURRED AT 05:48

LOW:  44 OCCURRED AT 05:34
HIGH: 89 OCCURRED AT 04:45

LOW:  44 OCCURRED AT 08:35
HIGH: 89 OCCURRED AT 04:45

LOW:  22 OCCURRED AT 02:21
HIGH: 22 OCCURRED AT 19:10
```

pr62 - Mowing Fields

Problem:	Write a program that will determine the length of time it will take to mow a rectangular field when it takes an average of one minute to mow a strip 2 feet wide and 45 feet long.
Input:	There is one line of input. The first integer on the line is the number of pairs of integers to follow. The remaining pairs 0f integers indicate the length and width of the field to be mowed in feet. The mower is 24 inches wide.
data file:	`pr62`
Output:	Output the number of hours and minutes in the format: hh:mm where hh is hours without leading zeros and mm is minutes with leading zeros if needed.
Assumptions:	None.
Sample Input:	`3 70 175 93 225 145 290`
Output Input:	

```
2:16
3:53
7:47
```

pr63 - Cell Phone 1

Problem:	Write a program that will print the total number of minutes used by a cell phone caller in a day.
Input:	The first line of the data set is an integer that represents the number of phone calls that follow. Each of the remaining lines contains the start and stop time of each call during the 24 hour period. All times are given in military time without leading zeros on the minutes and with leading zeros on the seconds (e.g. 2:05 PM would be 14:05)
data file:	`pr63`
Output:	Output the total talk time.
Assumptions:	None.

Sample Input:	`6`
	`8:22 8:35`
	`9:08 10:03`
	`11:46 12:23`
	`14:50 17:30`
	`21:21 22:29`
	`23:04 23:48`

Output Input:	`TOTAL MINUTES: 377`

pr64 - Duct Tape

Problem: Write a program that will determine the number of rolls of duct tape a person would need to buy and the total cost of the tape. A roll of duct tape contains 100 feet and the cost is cheaper when more rolls are bought as given by this chart:

1 roll	$4.95 per roll
6 rolls	$4.65 per roll
12 rolls	$4.15 per roll
24 rolls	$4.00 per roll

Input: The first line of the data set is an integer that represents the number of lines that follow. Each of the remaining lines contains the number of inches of duct tape needed.

data file: `pr64`

Output: Output the total cost and the number of rolls needed in the format shown below. Money needs to be rounded to the nearest penny and contain a dollar sign($).

Assumptions: Input values will be in the range 1 .. 30000.

Sample Input:

```
4
19059
4008
29999
13000
```

Output Input:

```
$66.40 FOR 16 ROLLS
$19.80 FOR 4 ROLLS
$100.00 FOR 25 ROLLS
$51.15 FOR 11 ROLLS
```

pr65 - Compound Interest

Problem: Write a program that will print the amount of money a person has in their savings account at the end of each month. The interest if compounded monthly, which means that the interest paid in month 1 earns interest in month 2, etc.

Input: The first line of the data set is an integer that represents the number of test cases that follow. Each of the remaining lines contains the amount of principal, the yearly interest rate written as a percent, and the number of months for which to print the balance.

data file: `pr65`

Output: Output the month number and the balance at the end of each month beginning with month 0, the original balance. The balance should be rounded to the nearest penny and include a dollar sign ($). All output should be right justified with at least one space between the month number and the balance.

Assumptions: Each month, the interest rate will be 1/12 of the yearly interest rate. There should be at least one blank line between set of output

Sample Input:

```
2
1000 2.5 10
5025 3.25 14
```

Output Input:

```
 0  $   1000.00
 1  $   1002.08
 2  $   1004.17
 3  $   1006.26
 4  $   1008.36
 5  $   1010.46
 6  $   1012.57
 7  $   1014.67
 8  $   1016.79
 9  $   1018.91
10  $   1021.03

 0  $   5025.00
 1  $   5038.61
 2  $   5052.26
 3  $   5065.94
 4  $   5079.66
 5  $   5093.42
 6  $   5107.21
 7  $   5121.04
 8  $   5134.91
 9  $   5148.82
10  $   5162.76
11  $   5176.75
12  $   5190.77
13  $   5204.83
14  $   5218.92
```

pr66 - Cash Returns

Problem: Write a program that will print the number of each coin a person should give in change. The number of coins should be as few as possible. The customer will pay in dollar bills and the amount of change needed will always be 99 cents or less.

Input: The first line of the data set is an integer that represents the number of lines that follow. Each of the remaining lines contains the amount of one purchase.

data file: `pr66`

Output: Output the fewest number of quarters, dimes, nickels, and pennies that could be given in change for each purchase in the format given below (one space after the colons (:) and two spaces before the D, N, and P).

Assumptions: The amount tendered will be the next integral number of dollars greater than the purchase price so the amount of changes will always be less than a dollar. For example, if $6.21 is due, $7 will be paid.

Sample Input:

```
4
6.21
4.87
2.01
.57
```

Output Input:

```
Q: 3  D: 0  N: 0  P: 4
Q: 0  D: 1  N: 0  P: 3
Q: 3  D: 2  N: 0  P: 4
Q: 1  D: 1  N: 1  P: 3
```

pr67 - Fence Me In

Problem: Write a program that will print the cost of fencing that would be needed to fence a yard of the dimensions given. The cost of a six-foot panel of fencing is $19.95 and the price of a 4-foot gate is $29.95.

Input: The first line of the data set is an integer that represents the number of lines that follow. Each of the remaining lines contains three integers that represent the length and width of the yard to be fenced and the number of gates needed. All measurements are in feet.

data file: `pr67`

Output: Output the total cost of the fencing and gates. The cost should be rounded to the nearest penny.

Assumptions: None.

Sample Input:

```
3
50 60 3
25 39 2
75 123 4
```

Sample Output:

```
$10024.95
$3291.80
$30743.05
```

pr68 - Cell Phone 2

Problem: Write a program that will print the total number of minutes used by a cell phone caller in a day.

Input: The first line of the data set is an integer that represents the number of phone calls that follow. Each of the remaining lines contains the start and stop time of each call during the 24 hour period. All times are given without leading zeros on the minutes and with leading zeros on the seconds (e.g. 2:05 PM would be 14:05) followed by either AM or PM.

data file: `pr68`

Output: Output the total talk time.

Assumptions: None.

Sample Input:

```
6
8:22 AM 8:35 AM
9:08 AM 10:03 AM
11:46 AM 12:23 PM
2:50 PM 5:30 PM
9:21 PM 10:29 PM
11:04 PM 11:48 PM
```

Sample Output:

```
TOTAL MINUTES: 377
```

pr69 - Grocery Bill

Problem: Write a program that will determine the percent of a person's grocery bill was spent in the following departments: meat, produce, food (boxed, canned, etc.), drugs, and other (paper goods, toys, etc.).

Input file name: `pr69.dat`

Input: There are an unknown number of lines. Each line will contain an uppercase letter of the alphabet to indicate the type of item it is and the cost of the item. The letter codes are:

M for meat
P for produce
F for food (boxed, canned, etc.)
D for drugs
O for other (paper goods, toys, etc.)

Output: For each department, print the total spent in the department and department's percent of the total bill. Round all percents to 10ths. Use the format as shown in the sample output.

Sample Input:

```
M 9.87
P .76
O 9.87
F 3.69
P 4.23
F 3.26
D 9.99
F .56
F 8.75
M 5.44
P 3.66
D 15
O 5.25
F 9.23
D 10.50
P 5.78
```

Sample Output:

```
Total bill: 105.84
Meat:     $  15.31   14.5%
Produce:  $  14.43   13.6%
Food:     $  25.49   24.1%
Drugs:    $  35.49   33.5%
Other:    $  15.12   14.3%
```