Assignment 3:
Modifying and Printing an Invoice

Dr. William Kreahling

April 6, 2021

Objectives: Continue writing working shell scripts, using correct control flow statements, exits status. Useful tools **may** include: wc, cut, grep, test, expr, awk, for, tr, sed, sort, tbl, groff, latex. You may use arrays and functions, and *shopt*: shell optimizations in bash.

Instructions: Your assignment is to write two shell scripts (and any accompanying helper scripts) to add and print records from a text invoice.

1. insert.sh
2. print.sh

You may create additional shell scripts as needed to abstract out common operations. While you are allowed to invoke other shell scripts that you have written and the Linux utilities specified above, you are not allowed to invoke other executables (i.e. you cannot write your solution in a conventional programming language like C++ or Java). If you need to store information into a temporary file, then you can use the file *tmp.txt*. Be sure to delete the temp file before you exit your script.

1. **insert.sh**:

   The bash script called *insert.sh* will insert a record into a text invoice (as developed in assignment two). It accepts two arguments, first, the invoice name and second, an optional argument representing the category of records you wish to insert. The *insert.sh* script should then prompt the user for any missing records in the invoice corresponding to the category supplied as the second argument. If no second argument is supplied then the script will prompt the user to enter records for **every** missing record in the script, regardless of category.

   For example, if the user specified that they want to buy 7 'produce' items and there are only 3 'produce' records in the invoice, the script will prompt them for 4 more records for 'produce' items. The script should read in the values for item name, cost, and number of items to purchase and append these values in a new line at the current end of the invoice file. Finally, the script should print a message that "N new invoice records" have been added. Below is an example session of using the *insert.sh* script. Note the field values would be entered by the user.

   Given the following invoice

```
customer:Paul Johnson
address:546 Broadway, New York, NY

categories:Produce,Services,Homewares,Toiletries
items:3,1,2,1

produce: apples, 54, 5
produce: oranges, 1.10, 9
services: window washing, 25.99, 9
homewares: dishes, 4.99, 10
toiletries: shaving cream, 1.45, 3
```

Here is an example of running insert.sh

```
> insert.sh example.oso
Please enter the name of a Produce item > carombola
Please enter a price per unit of carombola > 1.19
Please enter the amount of carombola units to purchase > 3

Please enter the name of a Homewares item > wash cloths
Please enter a price per unit of wash cloths > 0.99
Please enter the amount of wash cloths units to purchase > 15
2 records added to "example.oso" invoice
```

```
> insert.sh example.oso
0 records added to "example.oso" invoice
```

and now the file contains:

```
customer:Paul Johnson
address:546 Broadway, New York, NY

categories:Produce,Services,Homewares,Toiletries
items:3,1,2,1

produce: apples, 54, 5
produce: oranges, 1.10, 9
services: window washing, 25.99, 9
homewares: dishes, 4.99, 10
toiletries: shaving cream, 1.45, 3
Produce: carombola, 1.19, 3
Homewares: wash cloths, 0.99, 15
```

Print an appropriate error message(s) and exit, when errors are encountered.

Given the following invoice

```
customer:Paul Johnson
address:546 Broadway, New York, NY

categories:Produce,Services,Homewares,Toiletries
items:3,1,2,1
```

Here is an example of running insert.sh

```
> insert.sh example2.oso snausages

snausages is not a valid category for this invoice
```

```
>insert.sh example2.oso services
```

```
Please enter the name of a services item > life coaching
Please enter a price per unit of life coaching > 45.99
Please enter the amount of life coaching units to purchase > 3

1 records added to "example2.oso" invoice
```

2. **print.sh**: Write a bash shell script called *print.sh* that prints a text invoice that has been sorted on a specified field. The usage is as follows:

```
print.sh <invoice filename> [-c]
```

The *invoice filename* is required and the *-c* is optional. You should check that the invoice represents a valid invoice.

If a flag is present, you should check that it is a valid flag

The invoice should be printed as a table and should be sorted in order of the items, per category unless the *-c* flag is present and then it should be sorted in ascending order of total cost.

You should then create a PDF or PS file called *tmp.pdf*/*tmp.ps* using groff or latex and view it using *evince* or *okular*. After displaying the table, you should remove the *tmp.tex*/*tmp.tr* and *tmp.pdf*/*tmp.ps* files, along with any other temporary files and exit.

Below is an example invocation of the *print.sh* script.

```
% print.sh example.oso
```

Your shell script should produce a table of the data that would appear as:

Paul Johnson 546 Broadway, New York, NY, 10019-8765				
Category	Item	Cost	Quantity	Total
homewares	dishes	4.99	10	49.90
Homewares	wash cloths	0.99	15	14.85
produce	apples	54.00	5	270.00
Produce	carombola	1.19	3	3.57
produce	oranges	1.10	9	9.90
services	window washing	25.99	9	233.91
toiletries	shaving cream	1.45	3	4.35

Details:

1. All scripts should print usage messages if an incorrect number of command line arguments are supplied.
2. Command line arguments must be passed into your scripts in the order they are listed in this handout. Thus everyone's script will accept the arguments in the same order and *should* work with one another.
3. All scripts should exit and print error messages if the files we need to create/read/modify do not have correct permissions to allow us the access needed. If a file does not have write permissions but we are not writing to it, that should **not** be reported or cause the script to fail.
4. Use correct style and indentation, comments, name, etc.
5. Your output should match mine as closely as possible.
6. You must name your scripts with the names supplied in this document.
7. The tests shown in this document do **not** cover all errors that could occur.
8. You must submit a text document named `tests.txt` with a list of tests that you performed on your scripts. A step by step list of command and output is desirable.
9. invoices should print out a total cost per item purchased in the last column for each row The total cost should not be appended to the invoice file, but should show up in the printout.

Submission: Submit your shell scripts by midnight, April 12th using handin on agora (assignment number 3).