

ECE 118 – LAB TEN

1. MaxBargain supermarkets has approached you to develop a billing software for their stores. The input is a list of products purchased by customers and you need to create a software to calculate the total bill with tax and discounts.

The format of input file “items.txt” is as follows:

```
Milk 1 3.75 20 N
Wine 2 10.99 0 Y
Butter 1 4.99 5 N
Flour 4 3.99 0 N
Salmon 3 12.99 0 Y
```

The input file is organized as, Product Name, Quantity purchased (number of units), price per unit, discount applicable and if the product is eligible for tax.

Create a file **“receipt.txt”** that shows the following output. Print the input line along with the additional information of, total price for the units purchased, price after discount, tax amount (use 7% if eligible for tax) and price after tax. At the end of the file, print the number of items, the total price, tax and total payable after tax. The dollar amount values should not display more than 2 digits after decimal.

```
Milk 1 3.75 20 N 3.75 2.95 0.00 2.95
Wine 2 10.99 0 Y 21.98 20.98 1.47 22.45
Butter 1 4.99 5 N 4.99 4.04 0.00 4.04
Flour 4 3.99 0 N 15.96 14.96 0.00 14.96
salmon 3 12.99 0 Y 38.97 37.97 2.66 40.63
ITEM-COUNT 11
AMOUNT $80.90
TAX $4.13
TOTAL-PAYABLE $85.03
```

2. This is the continuation of the previous **question**. Make the above program, to process multiple files.
 - a. Add a part where the program is interactive to accept the customer name and the file name from console input.
 - b. For example: if the user gives input customer name as : Alex, the program should read from input file Alex_items.txt and be create the output receipts file as Alex_receipts.txt
 - c. If the user says, he does not want to continue, create a report which displays a summarized information. (Display on console and also write to a file consolidated_report.txt)

```
Total No. of items sold :  
Total Revenue (in $)      :  
Total Sales tax payable (in $)  :  
Revenue after tax (in $)
```

3. A file has a list of numbers “numbers.txt”. Create a file “prime_frequency.txt”, where it lists the number of prime numbers between 1 and the number inclusive.

Example input numbers.txt

```
11  
100  
214  
54  
33  
35
```

For example if the input is 11, the prime numbers less than and including 11 are (2,3,5,7,11), so the output should be 5.

Output file should have:

```
5  
25  
47  
16  
11  
11
```

4. A file has a list of words “words.txt”. Write a program to count the frequency of letters of the alphabet (characters) in the file.

For example input file “words.txt”

```
Apple  
Balloon  
Cat  
Donkey  
Elephant
```

Create a file frequency_one.txt with output as:

```
A 1  
B 1  
C 1  
D 1
```

```
E 1
a 3
e 3
h 1
k 1
l 4
n 3
o 3
p 3
t 2
y 1
```

Next step, take the file `frequency_one.txt` as input and create `frequency_two.txt`, where you have the counts of characters case insensitive.

Remember: for this part you should use the output from the first section only i.e frequency_one.txt file not the words.txt file

```
A 4
B 1
C 1
D 1
E 4
H 1
K 1
L 4
N 3
O 3
P 3
T 2
Y 1
```

5. A file called "people.txt" contains many thousands of lines of data that look just like this:

```
19911213 Douglas Davies 70 170 NJ
19420117 Cory DeMilt 68 175 NY
19620606 Heironymous McTaggart 72 231 LA
19591106 Ian Williams 74 320 FL
19700620 Dante Shellduck 64 168 CT
...
```

The data on each line are a person's birth date, their first and last name, their height in inches, their weight in pounds, and the state that they live in.

1. Write a complete C++ program, even showing the correct #includes, that reads this file and performs the following tasks:
2. Create another file containing the data only for people who live on the West coast - their state will be one of CA, OR, or WA.
3. Count and print the total number of people in the original file.
4. Print both names of the shortest person in the original file.