Week 6 Bash Questions

Problem 1

Write the bash script to convert roman numerals to integer and display. Example `XI equals 11`

Symbols and corresponding integer values are given below.
'M'=1000
'D'=500
'C'=100
'L'=50
'X'=10
'V'=5
'I'=1

A number in Roman Numerals is a string of these symbols written in descending order (e.g. M first, followed by D, etc.).
However, subtractive notation is used as follows:

I placed before V or X represents 1 less, so 4 is IV (5-1=4) and 9 is IX (10-1=9).
X placed before L or C represents 10 less, so 40 is XL (50-10=40) and 90 is XC (100-10=90).
C placed before D or M represents 100 less, so 400 is CD (500-100) and 900 is CM (1000-100).
The maximum limit possible to represent in roman numerals is 3999.

Solution

```
#declaring the array with conversion data
declare -A arr;
arr['M']=1000;
arr['D']=500;
arr['C']=100;
arr['L']=50;
arr['X']=10;
arr['V']=5;
arr['I']=1;

roman=$(echo $1 | tr a-z A-Z)

#Intiialize
n=$(echo "${#roman} -1" | bc)
p=0
ans=0

#Conversion
for (( i=$n; i>=0; i-- ))
do
  value=${roman:$i:1}
```

```
  if [ ${arr[$value]} -ge $p ] ;
  then
    ans=$(($ans + ${arr[$value]}))
  else
    ans=$(($ans - ${arr[$value]}))
  fi

  p=${arr[$value]}
done

echo "$roman equals $ans"
```

Test Cases

1. MMMDCCXXIV 3724
2. MXXIV 1024
3. MMMDCCLXXXVII 3787
4. LVIII 58

Problem 2

While developing software it is a common practice to add a copyright statement at the beginning of each file.
Write a script to add the copyright statement given in the variable `copyright` and add it in the beginning of all (`.c`) C source files.

Profix

```
copyright="This file is copyrighted under the BSD licensing"
echo -e "include headers here\ninclude others here" > t1.C
echo -e "include this here\ninclude that here" > t2.C
echo -e "inlcude me here\ninclude me there" > t3.C
```

Solution

```
for i in *.C
do
  echo $copyright > temp.txt
  cat $i >> temp.txt
  cat temp.txt > $i
  rm temp.txt
done
```

Problem 3

In your current directory, you have multiple files with the format `sensorNAME_day_Month_year.txt`.
Example: `sensorA_22_Jan_2021.txt`.

Write a script to rename all such files in the above format to `Month_year.txt` format.
Example: `Jan_2021.txt`

Prefix

```
touch sensorA_25_Feb_2021.txt sensorB_15_Jun_2022.txt sensorD_06_Dec_2021.txt
```

Solution

```
for i in sensor*.txt
do
  j=`echo $i | egrep -o '[A-Z][a-z]{2}_.*'`
  mv $i $j
done
```

Problem 4

Sometimes differences between American English and British English can cause confusions. The file `input.txt` contains British English words.
Convert them to corresponding American English words and update the file. A file `dict.csv` contains Biritish English words in the first column and
corresponding american english words in the second column.

Prefix

```
echo -e "aluminium,aluminum\nlift,elevator\npavement,sidewalk" > dict.csv
echo -e " Please do not walk on the aluminium pavement" > input.txt
```

Solution

```
declare -A dict;

while read i;
do
  key=`echo $i | cut -d "," -f 1`
  value=`echo $i | cut -d "," -f 2`
  dict[$key]=$value
done< $1
input=`cat $2 | tr [A-Z] [a-z]`
for j in ${!dict[@]}
do
  val=${dict[$j]}
  input=${input//$j/$val}
done

echo $input > $2
```

Problem 5

You have multiple files containing details of the item specifications provided by different suppliers with the format `supplier_item_date.txt`.
Example `sup1_pencil_120122.txt`. Write a script to create a directory for each supplier and within which there is a directory for each item and copy
the files to the corresponding folders. The supplier names and item names are present in `suppliers.txt` and `items.txt` respectively.

Prefix

```
echo -e "sup1\nsup2\nsup3" > suppliers.txt
echo -e "pen\npaper\nstapler" > items.txt
touch sup1_pen_201121.txt sup2_pen_230421.txt
touch sup1_paper_101021.txt sup3_stapler_090422.txt
```

Solution

```
sup_list=`cat suppliers.txt`
item_list=`cat items.txt`
for i in $sup_list
do
  mkdir $i
  for j in $item_list
  do
    mkdir $i/$j
    f=`ls | grep $i | grep $j`
    if [ $f > 0 ];
    then
      cp $f $i/$j
    fi
  done
done
```