HW9_IS457_85

Dec Nov 6, 2018

The script was run with the following arguments:

./HW9_IS457_85.sh 5 unix Hw_9.txt 8

(1) (2pts). Check whether your input integer(Argument_1) is even or odd and print your result. (5 points)

echo "******** Q1 **********"**

Ans :

```
VAL=$1

VAL1=2

let VAL2=VAL%VAL1

if [ "$VAL2" -eq 0 ]; then
        echo "The given number $1 is even"
else
        echo "The given number $1 is odd"
fi
```

Output :

```
********** Q1 **********
The given number 5 is odd
```

(2) (2pts). Input a lowercase letter(Argument_2) and convert it to uppercase and print your result. (5 points) (Hint: tr)

echo "******** Q2 **********"**

Ans :

```
echo "To Uppercase: $2" | tr "[a-z]" "[A-Z]"
```

Output :

```
********** Q2 **********
TO UPPERCASE: UNIX
```

(3) (2pts). Convert the following phrase "CS 398/IS 457:INTRODUCTION TO DATA SCIENCE" into separate words, and put each word on its own line (ignoring space,'/' and ':'). (5 points)

The output looks like:

CS
398
IS
457
INTRODUCTION

TO
DATA
SCIENCE

echo "******** Q3 **********"**

Ans :

```
IN="CS 398/IS 457:INTRODUCTION TO DATA SCIENCE"
array=$(echo $IN | tr "[:space:]" "\n" | tr "[:punct:]" "\n")

for str in $array
do
        echo "$str"
done
```

Output :

```
********** Q3 **********
CS
398
IS
457
INTRODUCTION
TO
DATA
SCIENCE
```

(4) (2pts). Sort the answer in Q3 by descending order. (5 points)

The output would look like:

TO
SCIENCE
IS
INTRODUCTION
DATA
CS
457
398

echo "******** Q4 **********"**

Ans :

```
echo "${array[@]}" | sort -r
```

Output :

```
********** Q4 **********
TO
SCIENCE
IS
INTRODUCTION
DATA
CS
457
398
```

(5) (2pts). Count the lines of your text file(Argument_3). (5 points) (Hint: wc)

```
echo "********** Q5 **********"
echo "The number of lines in $3 is:"
```

Ans :

```
wc -l $3
```

Output :

```
********** Q5 **********
The number of lines in Hw_9.txt is:
19 Hw_9.txt
```

(6) (2pts). Count the frequency of a input word(Argument_2) in a text file(Argument_3), and print "The frequency of word ____ is ____ ". (5 points) (Hint: grep)

```
echo "********** Q6 **********"
echo "The frequency of word $2 is:"
```

Ans :

```
grep -o -i -w $2 $3 | wc -l
```

Output :

```
********** Q6 **********
The frequency of word unix is:
6
```

(7) (2pts). Print the number of unique words in the text file(Argument_3). (5 points) (Hint: uniq, sort)

```
echo "********** Q7 **********"
echo "The number of unique words in the text file:"
```

Ans :

```
cat $3 | tr "[:upper:]" "[:lower:]" | tr -s " " "\n" | tr -s "[:punct:]" "\n" | sort | uniq -u | wc -l
```

Output :

```
********** Q7 **********
The number of unique words in the text file:
130
```

(8) (2pts). Print the number of words that begin with the letter 'b' in the text file(Argument_3) (5 points).

```
echo "********** Q8 **********"
echo "The number of words that begin with letter 'b':"
```

Ans :

```
cat $3 | tr "[:upper:]" "[:lower:]" | tr -s " " "\n" | tr -s "[:punct:]" "\n" | grep -o -i '^b' | wc -l
```

Output :

```
************ Q8 ************
The number of words that begin with letter 'b':
9
```

(9) (2pts). Print top-k(Argument_4) and find the most frequent word and their frequencies. (5 points). (Hint: head)

echo "********** Q9 ************"**
echo "Top-$4 words are:"

Ans :

cat $3 | tr "[:upper:]" "[:lower:]" | tr -s " " "\n" | tr -s "[:punct:]" "\n" | sort | uniq -c | sort -n -r | head -n $4

Output :

```
************ Q9 ************
Top-8 words are:
     58 the
     20 and
     14 shell
     13 of
      9 user
      9 to
      9 a
      7 system
```

(10) (4pts). The dataset adult-income.csv provides some clean records of adults to predict whether income exceeds $50K/yr. For details, visit the UCI repository. Calculate how many categories are there in "workclss" (2nd column) and print your result. (5 points) (Hint: awk)

echo "********** Q10 ************"**

Ans :

cat adult-income.csv | tr '\r' '\n' | awk -F, '{print $2}' | sort | uniq -c

Output :

```
************ Q10 ************
   1836  ?
    960  Federal-gov
   2093  Local-gov
      7  Never-worked
  22696  Private
   1116  Self-emp-inc
   2541  Self-emp-not-inc
   1298  State-gov
     14  Without-pay
```

(11) (4pts). For your output in Q10, change the format of categories. Replace "-" with "_". (Hint: sed)

The output would look like:

1
Private 22696
Local-gov 2093
State-gov 1298
...

echo "******** Q11 **********"**

Ans :

cat adult-income.csv | tr '\r' '\n' | awk -F, '{print $2}' | sort | uniq -c | sed 's/-/_/g'

Output :

```
********** Q11 **********
   1836  ?
    960  Federal_gov
   2093  Local_gov
      7  Never_worked
  22696  Private
   1116  Self_emp_inc
   2541  Self_emp_not_inc
   1298  State_gov
     14  Without_pay
```