

```
1 #PS01
. #Kehsin Su Esther
. #3033114294
. #2
- file
=
"
http
:
//
data
.
un
.
org
/
Handlers
/
DownloadHandler
.
ashx
?
DataFilter
=
itemCode
:526&DataMartId=FAO&Format=csv&c=2,3,4,5,6,7&s=countryName:asc,elementCode:asc,year:desc"
. #get data from the internet
. curl -o data.zip $file
.
. #unzip the data
10 unzip -s data.zip
. #view the data
. less UNdata_Export_20170902_045616030.csv
.
. #rename the csv file
- mv UNdata_Export_20170902_045616030.csv apricots.csv
.
. #separe the data into two part by area and country
. grep -w '+' apricots.csv > apricots_area.csv
. grep -v '+' apricots.csv > apricots_country.csv
20
. #(a)
. grep data with specific string "2005" and "area_harvested", then remove " and sorted it
with number, at the 6th column with reverse ordered(r). Finally use echo and head to print
out the first five rows in the screen.
. grep 2005 apricots_country.csv | grep "Area Harvested" | sed 's/"//g' | sort -n -t ','
-k 6 -r | head -n 5
.
- #use a for loop to get the top five country that has the greatest value every 10 years
. for ((i=1965; i<=2005; i+=10)) ; do
.     grep $i apricots_country.csv | grep "Area Harvested" | sed 's/"//g' | sort -n -t ','
-k 6 -r | head -n 5
. done
.
30 #yes, the rankings have changed
.
. #(b)
. #write a function to grep data, if the user enter "-h", we should instruct he/she what this
function is used for, then if the size of the file is too small, we might guess it is no
data in the file, so we can inform the user the item code he/she is using is wrong.
.
- function getdata(){
.     if [ "$1" == "-h" ]
.     then
.         echo "input a number to get data"
.         exist 0;
40     fi
.     file
.     =
.     "
.     http
```

```

:
//
data
.
un
.
org
/
Handlers
/
DownloadHandler
.
ashx
?
DataFilter
=
itemCode
:
$1&DataMartId=FAO&Format=csv&c=2,3,4,5,6,7&s=countryName:asc,elementCode:asc,year:desc"
42 curl -o data.zip $file
. unzip -p data.zip > data.csv
.
- byte=$(ls -la data.csv | cut -d' ' -f5)
. if [ $byte -lt 1000 ]
. then
.     echo "error item code"
. fi
50 }
.
. #3
. #first download the html file
. #then from the file, we can grep those txt filenames and use these filenames in for loop to
continuous download them
- #use echo to print out the filename we are download
.
. url="https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/"
. wget $url
. text=$(grep .txt index.html | cut -d '"' -f8)
60 for i in $text; do
. echo $i
. wget https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/$i; done
.
.
-
.
```