

Learn the following for the OS lab

The directory structure of UNIX, absolute path, relative path, hidden files

Special characters of the UNIX shell---wild card characters---*, ?, [a-z], ;
etc...

Terminal control keys---CTRL-d, CTRL-h, etc...

Commands---echo, cmp, diff, wc, grep, tail, od, file, du,, sed,awk

Shell programming----positional parameters and arguments--for

The tasks for the lab session

1. Create a text file sourcefile using an editor and count the number of characters, words and lines in it.

```
auist@auist-Veriton-M200-H81:~/Documents/5013/dir1/sub_dir1$ wc
dec22_file1_2009..2022115013
 5  30 177 dec22_file1_2009..2022115013
```

2. Display only the last 3 lines of the above file.

```
auist@auist-Veriton-M200-H81:~/Documents/5013/dir1/sub_dir1$ tail -n 3
dec22_file1_2009..2022115013
Linux is a great os.Linux was developed in Bell labs.
Linux is easy to learn.Linux is a multiuser os.Learn linux.
Linux is powerful.
```

3. Copy the above file into another destinationfile and edit the second file by inserting a few more words, lines in it.

```
auist@auist-Veriton-M200-H81:~/Documents/5013$ cp
dir1/sub_dir1/dec22_file1_2009..2022115013 dir3
auist@auist-Veriton-M200-H81:~/Documents/5013$ printf "Have a good day.">>
dir3/dec22_file1_2009..2022115013
auist@auist-Veriton-M200-H81:~/Documents/5013$ cat
dir3/dec22_file1_2009..2022115013
Hello World!!
I am learning linux commands.
Linux is a great os.Linux was developed in Bell labs.
Linux is easy to learn.Linux is a multiuser os.Learn linux.
Linux is powerful.
Have a good day.
```

4. Compare sourcefile with destinationfile using cmp and diff and note the difference between the two commands.

```
auist@auist-Veriton-M200-H81:~/Documents/5013$ cmp
dir1/sub_dir1/dec22_file1_2009..2022115013 dir3/dec22_file1_2009..2022115013
cmp: EOF on dir1/sub_dir1/dec22_file1_2009..2022115013 after byte 177, line
5
auist@auist-Veriton-M200-H81:~/Documents/5013$ diff
dir1/sub_dir1/dec22_file1_2009..2022115013 dir3/dec22_file1_2009..2022115013
5a6
> Have a good day.
\ No newline at end of file
```

5. Display the lines having a specific word in sourcefile

```
auist@auist-Veriton-M200-H81:~/Documents/5013$ grep "Linux"
dir1/sub_dir1/dec22_file1_2009..2022115013
Linux is a great os.Linux was developed in Bell labs.
Linux is easy to learn.Linux is a multiuser os.Learn linux.
Linux is powerful.
```

6. Display the lines not having a specific word in sourcefile

```
auist@auist-Veriton-M200-H81:~/Documents/5013$ grep -v "Linux"
dir1/sub_dir1/dec22_file1_2009..2022115013
Hello World!!
I am learning linux commands.
```

7. How to go back to your home directory from any other place in the directory structure?

```
auist@auist-Veriton-M200-H81:~/Documents/5013$ cd
```

8. How to retrieve the previous Shell commands that you had used in the current log-in session?

```
auist@auist-Veriton-M200-H81:~$ history
```

9. Display the calendar of a particular month.

```
auist@auist-Veriton-M200-H81:~$ cal 02 2024
February 2024
Su Mo Tu We Th Fr Sa
      1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29
```

10. Check how od works on sourcefile.

```
auist@auist-Veriton-M200-H81:~/Documents$ od
5013/dir1/sub_dir1/dec22_file1_2009..2022115013
00000000 062510 066154 020157 067527 066162 020544 005041 020111
00000020 066541 066040 060545 067162 067151 020147 064554 072556
00000040 020170 067543 066555 067141 071544 005056 064514 072556
00000060 020170 071551 060440 063440 062562 072141 067440 027163
00000100 064514 072556 020170 060567 020163 062544 062566 067554
00000120 062560 020144 067151 041040 066145 020154 060554 071542
00000140 005056 064514 072556 020170 071551 062440 071541 020171
00000160 067564 066040 060545 067162 046056 067151 074165 064440
00000200 020163 020141 072555 072154 072551 062563 020162 071557
00000220 046056 060545 067162 066040 067151 074165 005056 064514
00000240 072556 020170 071551 070040 073557 071145 072546 027154
00000260 000012
00000261
```

11. Create a executable using an editor which the following shell commands in sequence: Who, sort, calendar, cat, grep with appropriate parameters / arguments (for example cat requires a file name & !)

```
auist@auist-Veriton-M200-H81:~/Documents/5013$ nano mycommand.sh
#!/bin/bash

echo "who"
who
echo "sort"
sort dir1/sub_dir1/dec22_file1_2009..2022115013
echo "cal"
cal 02 2024
echo "cat"
cat dir1/sub_dir1/dec22_file1_2009..2022115013
echo "grep"
grep "Linux" dir1/sub_dir1/dec22_file1_2009..2022115013
auist@auist-Veriton-M200-H81:~/Documents/5013$ chmod +x mycommand.sh
auist@auist-Veriton-M200-H81:~/Documents/5013$ ./mycommand.sh
who
auist      :0                2023-03-02 18:29 (:0)
sort
Hello World!!
I am learning linux commands.
Linux is a great os.Linux was developed in Bell labs.
Linux is easy to learn.Linux is a multiuser os.Learn linux.
Linux is powerful.
cal
      February 2024
Su Mo Tu We Th Fr Sa
                1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29

cat
Hello World!!
I am learning linux commands.
Linux is a great os.Linux was developed in Bell labs.
Linux is easy to learn.Linux is a multiuser os.Learn linux.
Linux is powerful.
grep
Linux is a great os.Linux was developed in Bell labs.
Linux is easy to learn.Linux is a multiuser os.Learn linux.
Linux is powerful.
```

12. Find out the type of executable.

```
auist@auist-Veriton-M200-H81:~/Documents/5013$ file mycommand.sh
mycommand.sh: Bourne-Again shell script, ASCII text executable
```

13. Make executable an executable file and execute it.

```
auist@auist-Veriton-M200-H81:~/Documents/5013$ chmod +x mycommand.sh
auist@auist-Veriton-M200-H81:~/Documents/5013$ ./mycommand.sh
```

14. Check how od works on executable.

```

auist@auist-Veriton-M200-H81:~/Documents/5013$ od mycommand.sh
00000000 020443 061057 067151 061057 071541 005150 062412 064143
00000020 020157 073442 067550 005042 064167 005157 061545 067550
00000040 021040 067563 072162 005042 067563 072162 020040 064544
00000060 030562 071457 061165 062137 071151 027461 062544 031143
00000100 057462 064546 062554 057461 030062 034460 027056 030062
00000120 031062 030461 030065 031461 062412 064143 020157 061442
00000140 066141 005042 060543 020154 031060 031040 031060 005064
00000160 061545 067550 021040 060543 021164 061412 072141 020040
00000200 064544 030562 071457 061165 062137 071151 027461 062544
00000220 031143 057462 064546 062554 057461 030062 034460 027056
00000240 030062 031062 030461 030065 031461 062412 064143 020157
00000260 063442 062562 021160 063412 062562 020160 046042 067151
00000300 074165 020042 064544 030562 071457 061165 062137 071151
00000320 027461 062544 031143 057462 064546 062554 057461 030062
00000340 034460 027056 030062 031062 030461 030065 031461 000012
00000357

```

15. Write a shell script which takes a sequence of filenames as arguments and does the following on each of them:
 Display the contents of the file; Count the number of characters, words, and lines in it; Display only those lines that contains the word 'the' in it; Display only the last 5 lines of it.

```

auist@auist-Veriton-M200-H81:~/Documents/5013$ nano file_info.sh

#!/bin/bash
for file in "$@"; do
    echo "File: $file"
    echo "----- Contents -----"
    cat "$file"
    echo "----- Word, Line, and Character Count -----"
    wc "$file"
    echo "----- Lines Containing the Word 'the' -----"
    grep "the" "$file"
    echo "----- Last 5 Lines -----"
    tail -n 5 "$file"
    echo "-----"
done
auist@auist-Veriton-M200-H81:~/Documents/5013$ chmod +x file_info.sh
auist@auist-Veriton-M200-H81:~/Documents/5013$ ./file_info.sh
dir1/sub_dir1/dec22_file1_2009..2022115013
File: dir1/sub_dir1/dec22_file1_2009..2022115013
----- Contents -----
Hello World!!
I am learning linux commands.
Linux is a great os.Linux was developed in Bell labs.
Linux is easy to learn.Linux is a multiuser os.Learn linux.
Linux is powerful.
----- Word, Line, and Character Count -----
  5  30 177 dir1/sub_dir1/dec22_file1_2009..2022115013
----- Lines Containing the Word 'the' -----
----- Last 5 Lines -----
Hello World!!
I am learning linux commands.
Linux is a great os.Linux was developed in Bell labs.
Linux is easy to learn.Linux is a multiuser os.Learn linux.
Linux is powerful.

```

16. Refine the above script to accept the word to be looked for and the number of lines to be displayed.

```
auist@auist-Veriton-M200-H81:~/Documents/5013$ nano file_info.sh
```

```
#!/bin/bash
for file in "${a:3}"; do
    echo "File: $file"
    echo "----- Contents -----"
    cat "$file"
    echo "----- Word, Line, and Character Count -----"
    wc "$file"
    echo "----- Lines Containing the Word '$1' -----"
    grep "$1" "$file"
    echo "----- Last 5 Lines -----"
    tail -n 5 "$file"
    echo "-----"
done
auist@auist-Veriton-M200-H81:~/Documents/5013$ chmod +x file_info.sh
auist@auist-Veriton-M200-H81:~/Documents/5013$ ./file_info.sh
dir1/sub_dir1/dec22_file1_2009..2022115013 the
File: dir1/sub_dir1/dec22_file1_2009..2022115013
----- Contents -----
Hello World!!
I am learning linux commands.
Linux is a great os.Linux was developed in Bell labs.
Linux is easy to learn.Linux is a multiuser os.Learn linux.
Linux is powerful.
----- Word, Line, and Character Count -----
   5  30 177 dir1/sub_dir1/dec22_file1_2009..2022115013
----- Lines Containing the Word 'the' -----
----- Last 5 Lines -----
Hello World!!
I am learning linux commands.
Linux is a great os.Linux was developed in Bell labs.
Linux is easy to learn.Linux is a multiuser os.Learn linux.
Linux is powerful.
```

17. Modify the above script to also display the first \hat{a} \hat{n} lines of any file.

```
auist@auist-Veriton-M200-H81:~/Documents/5013$ nano file_info.sh
```

```
#!/bin/bash
for file in "${a:4}"; do
    echo "File: $file"
    echo "----- Contents -----"
    cat "$file"
    echo "----- Word, Line, and Character Count -----"
    wc "$file"
    echo "----- Lines Containing the Word '$1' -----"
    grep "$1" "$file"
    echo "----- Last '$2' Lines -----"
    tail -n "$2" "$file"
    echo "----- First '$3' Lines -----"
    head -n "$3" "$file"
    echo "-----"
done
auist@auist-Veriton-M200-H81:~/Documents/5013$ chmod +x file_info.sh
auist@auist-Veriton-M200-H81:~/Documents/5013$ ./file_info.sh
dir1/sub_dir1/dec22_file1_2009..2022115013 the 5 2
```

```

File: dir1/sub_dir1/dec22_file1_2009..2022115013
----- Contents -----
Hello World!!
I am learning linux commands.
Linux is a great os.Linux was developed in Bell labs.
Linux is easy to learn.Linux is a multiuser os.Learn linux.
Linux is powerful.
----- Word, Line, and Character Count -----
   5  30 177 dir1/sub_dir1/dec22_file1_2009..2022115013
----- Lines Containing the Word 'the' -----
----- Last 5 Lines -----
Hello World!!
I am learning linux commands.
Linux is a great os.Linux was developed in Bell labs.
Linux is easy to learn.Linux is a multiuser os.Learn linux.
Linux is powerful.
----- First 2 Lines -----
Hello World!!
I am learning linux commands.

```

Grep Exercises.

1) Create an employee database (.txt file) which contains employee personal record. The personal record includes name, DOB (DD/MM/YYYY), address, phone number.

```

5013.txt
John Doe, 15/06/1985, 123 Main St, 555-1234
Jane Smith, 02/11/1979, 456 Oak St, 555-5678x101
Kumar Patel, 23/09/1988, 789 Pine St, 555-8765
Anita Singh, 30/05/1990, 321 Birch St, 555-4321x202
Raj Kumar, 08/03/1982, 654 Cedar St, 555-9876
Alan Brown, 19/04/1995, 321 Oak St, 555-5432
Mia Johnson, 12/01/1993, 789 Maple St, 555-6789
Alex Turner, 25/12/1975, 123 Spruce St, 555-3456

```

a) Print all Employee record that contain a phone number with an Extension

```

grep -E 'x[0-9]+' 5013.txt

Jane Smith, 02/11/1979, 456 Oak St, 555-5678x101
Anita Singh, 30/05/1990, 321 Birch St, 555-4321x202

```

b) Print all Employee record that do not begin with a capital A.

```

grep -v '^A' 5013.txt

John Doe, 15/06/1985, 123 Main St, 555-1234
Jane Smith, 02/11/1979, 456 Oak St, 555-5678x101
Kumar Patel, 23/09/1988, 789 Pine St, 555-8765
Raj Kumar, 08/03/1982, 654 Cedar St, 555-9876
Mia Johnson, 12/01/1993, 789 Maple St, 555-6789

```

c) Print all Employee record whose name ends with Kumar.

```

grep 'Kumar$' 5013.txt

Kumar Patel, 23/09/1988, 789 Pine St, 555-8765
Raj Kumar, 08/03/1982, 654 Cedar St, 555-9876

```

d) Print all the Employee names

```
awk -F', ' '{print $1}' 5013.txt
```

```
John Doe
Jane Smith
Kumar Patel
Anita Singh
Raj Kumar
Alan Brown
Mia Johnson
Alex Turner
```

e) Print all the Mobile number and the land line number with name

```
awk -F', ' '{print $1 ": " $4}' 5013.txt
```

```
John Doe: 555-1234
Jane Smith: 555-5678x101
Kumar Patel: 555-8765
Anita Singh: 555-4321x202
Raj Kumar: 555-9876
Alan Brown: 555-5432
Mia Johnson: 555-6789
Alex Turner: 555-3456
```

f) Find the eldest and the youngest employee from the record set.

```
sort -t', ' -k2,2 5013.txt | head -1
sort -t', ' -k2,2 5013.txt | tail -1
```

```
Alex Turner, 25/12/1975, 123 Spruce St, 555-3456
Alan Brown, 19/04/1995, 321 Oak St, 555-5432
```

g) Find the age of all the employees.

```
age.sh
#!/bin/bash
today=$(date '+%Y-%m-%d')
```

```
awk -F', ' -v today="$today" '{split($2, dob, "/");
dob_str = dob[3] "-" dob[2] "-" dob[1];
age = int((strftime("%Y%m%d", mktime(sprintf("%d %d %d 0 0 0", dob[3],
dob[2], dob[1])))) - strftime("%Y%m%d", mktime(sprintf("%Y %m %d 0 0 0",
substr(today, 1, 4), substr(today, 6, 2), substr(today, 9, 2))))) / 10000);
print $1 ": " age " years"
}' 5013.txt
```

```
John Doe: 38 years
Jane Smith: 44 years
Kumar Patel: 35 years
Anita Singh: 33 years
Raj Kumar: 42 years
Alan Brown: 29 years
Mia Johnson: 31 years
Alex Turner: 48 years
```

2) Create book database (.txt file) which contains book name, author name,

price,
publisher name and total page numbers.

book.txt

Database Management Systems, Abraham Silberschatz, 50, Pearson, 800
Introduction to Databases, C.J. Date, 45, Addison-Wesley, 600
SQL in 10 Minutes, Sams Teach Yourself, Ben Forta, 20, Sams Publishing, 200
NoSQL Distilled, Martin Fowler, 30, Addison-Wesley, 300
Python Programming, John Zelle, 35, Franklin, Beedle & Associates Inc., 400
Java: A Beginner's Guide, Herbert Schildt, 40, McGraw-Hill, 500
C++ Primer, Stanley B. Lippman, 45, Addison-Wesley, 600
JavaScript: The Good Parts, Douglas Crockford, 25, O'Reilly Media, 250

a) Change the publisher name from Mc Graw Hill to Tata Mc Graw Hill

sed -i 's/McGraw-Hill/Tata McGraw-Hill/' book.txt

b) Find the list of books which are related with the keyword "database"

grep -i 'database' book.txt

Database Management Systems, Abraham Silberschatz, 50, Pearson, 800
Introduction to Databases, C.J. Date, 45, Addison-Wesley, 600

c) Find the price of all programming language books

grep -E 'Python|Java|C\+\+' book.txt | awk -F', ' '{print \$1, ":", \$3}'

Python Programming : 35
Java: A Beginner's Guide : 40
C++ Primer : 45

d) Sort the books based on year of publication

sort -t, -k5,5n book.txt

SQL in 10 Minutes, Sams Teach Yourself, Ben Forta, 20, Sams Publishing,
200
JavaScript: The Good Parts, Douglas Crockford, 25, O'Reilly Media, 250
NoSQL Distilled, Martin Fowler, 30, Addison-Wesley, 300
Python Programming, John Zelle, 35, Franklin, Beedle & Associates Inc.,
400
Java: A Beginner's Guide, Herbert Schildt, 40, Tata McGraw-Hill, 500
C++ Primer, Stanley B. Lippman, 45, Addison-Wesley, 600
Database Management Systems, Abraham Silberschatz, 50, Pearson, 800

e) delete the low price book details

grep -E '([^\n]*,[^,]+,[3-9][0-9])|([^\n]*,[^,]+,[1-2][5-9])' book.txt >
temp.txt && mv temp.txt book.txt

Database Management Systems, Abraham Silberschatz, 50, Pearson, 800
Introduction to Databases, C.J. Date, 45, Addison-Wesley, 600
NoSQL Distilled, Martin Fowler, 30, Addison-Wesley, 300
Python Programming, John Zelle, 35, Franklin, Beedle & Associates Inc.,
400
Java: A Beginner's Guide, Herbert Schildt, 40, Tata McGraw-Hill, 500
C++ Primer, Stanley B. Lippman, 45, Addison-Wesley, 600
JavaScript: The Good Parts, Douglas Crockford, 25, O'Reilly Media, 250

3) Create a text file that contains the following strings

Aggregate, segregate, logicgate, certificate, appropriate, accumulate, accurate, acetate, adequate, anticipate, birthdate, borate, corporate, celebrate, complicate, chocolate, illustrate, affiliate.

1) Print all the record that starts with `^a` and ends with `ate`

```
grep -o '\ba[^,]*ate\b' example.txt
```

```
Aggregate
appropriate
accumulate
accurate
acetate
adequate
anticipate
```

2) Replace `birthdate` with `birthrate` and update in the source text file.

```
sed -i 's/birthdate/birthrate/' example.txt
```

3) print the record that ends `rate`

```
grep -o '\b[^,]*rate\b' example.txt
```

```
borate
celebrate
accurate
birthrate
```

4) Display the records with the pattern (gg) (cc) (bb) (ll) (ff)

```
grep -o '\bw*(gg|cc|bb|ll|ff)\w*\b' example.txt
```

```
Aggregate
appropriate
accumulate
accurate
illustrate
affiliate
```

5) Display the records that does not contains the pattern `gate`

```
grep -o '\b[^,]*\b' example.txt | grep -v 'gate'
```

```
certificate
appropriate
accumulate
accurate
acetate
adequate
anticipate
birthrate
borate
corporate
celebrate
complicate
chocolate
illustrate
affiliate
```

4) SED exercise

a) Write a sed command that will go through a file and eliminate

any .5 at the end of a record. If .5 is anywhere else in the record, leave it alone. Have the sed command only display those records which are modified.

Sample input :

```
1:3:5:7.5
1.5:2.5:3.5:7.5
1.5:2:3:4
1:2.5:3:4
```

```
sed -n 's/\.5$//p' input_file.txt
```

```
1:3:5:7
1.5:2.5:3.5:7
```

b) Given a data file where each record contains four fields and each field is separated from the others by a colon (:), write an awk command which will display the records with fields 3 and 4 swapped.

Input:

```
CIS:160:374:A
CIS:170:373:A
CIS:118:374:A
CIS:111:374:A
CIS:150:375:A
CIS:123:350:A
```

```
awk -F':' '{print $1 ":" $2 ":" $4 ":" $3}' input_file.txt
```

```
CIS:160:A:374
CIS:170:A:373
CIS:118:A:374
CIS:111:A:374
CIS:150:A:375
CIS:123:A:350
```

c) Use grep to find and display all lines in a file which contain either the string dog or the string cat.

Input: It's been quite a day.
It started when the cat woke me up.
The day was dreary. The weather
dogged me all day long.

```
grep -E 'dog|cat' input_file.txt
```

```
It started when the cat woke me up.
dogged me all day long.
```

d) This section turns a paragraph of text into something similar to leet speak.

Write a tr command that will turn all uppercase characters in a file to lowercase. Send the output from that command to a sed script which does the following:

```
convert the string ck to x
convert the string xs to xor
convert the string er to or
convert the string elite to leet
convert the character o to 0
```

```

convert the character t to 7
convert the character e to 3
convert the character s to 5
convert the character i to 1
convert the character a to 4

```

Input:

The hacker elite were gathering for their annual meeting. DEFCON had become an annual event. The big change this year was the appearance of more female hackers, who seemed to prefer being known as chicks. The one thing everyone seemed to agree on was that Windoze security sucks.

```

tr '[:upper:]' '[:lower:]' < input_file.txt | sed \ -e 's/ck/x/g' \ -e
's/xs/xor/g' \ -e 's/er/or/g' \ -e 's/elite/leet/g' \ -e 's/o/0/g' \ -e
's/t/7/g' \ -e 's/e/3/g' \ -e 's/i/1/g' \ -e 's/a/4/g'

```

```

7h3 h4x0r l33t w0r3 g47h3r1ng f0r 7h31r
4nnu4l m33t1ng. d3fc0n h4d b3c0m3 4n 4nnu4l
3v3n7. 7h3 b1g ch4ng3 7h15 y34r w45 7h3
4pp34r4nc3 0f m0r3 f3m4l3 h4x0r5, wh0 s33m3d
70 pr3f0r b31ng kn0wn 45 ch1x5. 7h3 0n3
7h1ng 3v3ry0n3 s33m3d 70 4gr33 0n w45 7h47
w1nd0z3 s3cur17y 5uck5.

```

Awk Exercise:

1) Create a file with the following fields: country name, capital, area, population and continent.

countries.txt

```

India,New Delhi,3287263,1393409038,Asia
United States,Washington D.C.,9833517,331002651,North America
Brazil,Brasilia,8515767,212559417,South America
Nigeria,Abuja,923768,206139589,Africa
Russia,Moscow,17098242,144104080,Europe
Australia,Canberra,7692024,25687041,Oceania
China,Beijing,9596961,1439323776,Asia
Japan,Tokyo,377973,126476461,Asia
Egypt,Cairo,1002450,102334404,Africa
South Africa,Pretoria,1219090,59308690,Africa

```

a) use awk and print the record that containing ~asia~™ and africa~™

```

awk -F',' ' /Asia|Africa/ {print}' countries.txt

```

```

India,New Delhi,3287263,1393409038,Asia
China,Beijing,9596961,1439323776,Asia
Japan,Tokyo,377973,126476461,Asia
Nigeria,Abuja,923768,206139589,Africa
Egypt,Cairo,1002450,102334404,Africa
South Africa,Pretoria,1219090,59308690,Africa

```

b) display the rtotal number of input record and fields for the above file.

```

awk -F',' ' BEGIN {count = 0; field_count = 0} {count++; field_count = NF}
END {print "Total records:", count; print "Total fields per record:",
field_count}' countries.txt

```

Total records: 10
Total fields per record: 5

c) set the field separator as tab (\t). display the sum of area and population for all the records.

```
awk -F',' '{area_sum += $3; population_sum += $4} END {print "Total area:", area_sum, "\nTotal population:", population_sum}' countries.txt
```

Total area: 61198609
Total population: 4504106096

d) select all the record that begins with letter â ~sâ™ through â ~zâ™

```
awk -F',' 'tolower($1) ~ /^[s-z]/ {print}' countries.txt
```

United States,Washington D.C.,9833517,331002651,North America
South Africa,Pretoria,1219090,59308690,Africa

e) select and display all the records where the population is in 5 digit number.

```
awk -F',' '$4 >= 10000 && $4 < 100000 {print}' countries.txt
```

Australia,Canberra,7692024,25687041,Oceania