

Quiz navigation

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

Finish review

Started on Wednesday, August 3, 2022, 6:53 AM
State Finished
Completed on Wednesday, August 3, 2022, 7:01 AM
Time taken 7 mins 43 secs
Marks 25.00/25.00
Grade 100.00 out of 100.00

Question 1

Correct

1.00 points out of 1.00

Flag question

Scientists who use UNIX often deal with data that are stored in text files. A primary need for these users is to extract particular portions of that database or to manipulate values in the database. In essence, one needs a program which permits easy manipulation of structured data and the generation of formatted reports. The `awk` program is one of the main tools for performing these actions.

The `awk` utility is a pattern scanning and processing program. It searches one or more files to see if they contain lines that match specified **patterns** and then perform associated **actions**, such as writing the line to the standard output or incrementing a counter each time it finds a match. The capabilities of `awk` extend the idea of text editing into computation, making it possible to perform a variety of data processing tasks, including analysis, extraction, and reporting of data.

The `awk` program has the following command line syntax

```
(file) jqwqecr&ltlt[your_username]> awk options program variables filenames
```


When using `awk`, we have to tell it which data we wish to process (i.e., the filename) and then how we wish to process it (the program instructions). The program instruction can be specified on the command line if it is enclosed by apostrophe marks (i.e., '{print "me"}') or they can be called from a file. For this course, we will use simple enough programs that we can just specify the instructions on the command line. The other options and variables are not required by `awk`, but we may use them later on in the course.

An `awk` program consists of one or more program lines containing a pattern and/or action in the following format:

```
pattern { action }
```


The **pattern** selects lines from the input file. The `awk` utility performs the action on all lines that the pattern selects. You must enclose the **action** within braces so that `awk` can differentiate it from the pattern.
Since there are several different components to using `awk` on the command line, let's make sure we can recall how the pieces fit together. What is the order of the following components when you write an `awk` command?

pattern3

variable5

filename6

awk1

action4

option2

The correct answer is: pattern – 3, variable – 5, filename – 6, awk – 1, action – 4, option – 2
Correct
Marks for this submission: 1.00/1.00.

Question 2

Correct

1.00 points out of 1.00

Flag question

To start, `awk` compares the first line in the input file with each pattern in the program. If a pattern selects a line (if there is a match), `awk` takes the action associated with the pattern. If the line is not selected, `awk` takes no action. When `awk` has completed its comparisons for the first line of the input file, it repeats the process for the next line of input. It continues this process, comparing subsequent lines in the input file, until it has read the entire input file's.
NOTE: There are two rules which occur if either a pattern or action is omitted. If a program line does not contain a pattern, `awk` selects all lines in the input file. If the program line does not contain an action, `awk` copies the selected lines to its standard output (this is usually the display, if you haven't redirected the output to another program or to a file).
Which of the following statements are correct about how `awk` behaves?

- Select one or more:
- ☐ a. Not including an action statement will ignore all lines that match the pattern
 - ☐ b. Not including an action statement will print all lines that do not match the pattern
 - ☐ c. Not including a pattern statement will perform the action on none of the lines
 - ☒ d. Not including a pattern statement will perform the action on all lines ✓ 1 of 2 correct answers
 - ☒ e. Not including an action statement will print all lines that match the pattern ✓ 1 of 2 correct answers

The correct answer is: Not including a pattern statement will perform the action on all lines, Not including an action statement will print all lines that match the pattern
Correct
Marks for this submission: 1.00/1.00.

Question 3

Correct

1.00 points out of 1.00

Flag question

To begin using `awk`, we need to have a file that we can use as our text database. First, we need to create a new directory **act3** in your **groupwork** directory where we can work on a text file with `awk`. After logging into OpenSARlab (OSL) and launching the desktop server and Linux terminal, starting from your home directory, you need to create the **act3** directory and move into the new directory. Which of the following commands would be needed to accomplish this?

- Select one or more:
- ☐ a. make act3
 - ☒ b. cd groupwork/act3 ✓ 1 of 2 correct answers
 - ☒ c. mkdir groupwork/act3 ✓ 1 of 2 correct answers
 - ☐ d. mv act3

The correct answer is: cd groupwork/act3, mkdir groupwork/act3
Correct
Marks for this submission: 1.00/1.00.

Question 4

Correct

1.00 points out of 1.00

Flag question

We will use a short text database of earthquakes to help demonstrate how `awk` works. We need to copy the `eqks.txt` file from `/home/jovyan/iris_data/SSBWFiles/eqks.txt` to your **act3** directory. If you completed the commands in the previous question correctly, you should be in the `/home/jovyan/groupwork/act3` directory (you can use `pwd` to check this). Which of the following commands would successfully copy the file to the **act3** directory?

- Select one:
- ☐ a. cp eqks.txt
 - ☐ b. cp /home/jovyan/iris_data/SSBWFiles/eqks.txt
 - ☐ c. cp /home/jovyan/iris_data/SSBWFiles/eqks.txt ~
 - ☐ d. cp eqks.txt .
 - ☐ e. cp eqks.txt act3
 - ☒ f. cp /home/jovyan/iris_data/SSBWFiles/eqks.txt . ✓

The correct answer is: cp /home/jovyan/iris_data/SSBWFiles/eqks.txt .
Correct
Marks for this submission: 1.00/1.00.

Question 5

Correct

1.00 points out of 1.00

Flag question

Using `cat` or `more` you should be able to see that the file looks like this

```
year month day hour minute second latitude longitude depth mag-type magnitude
04 12 26 00 58 53.0 03.287 95.972 30.0 MW 9.0
04 12 26 01 17 10.0 04.999 94.305 30.0 MB 5.5
04 12 26 01 21 21.0 06.340 93.342 35.1 MB 6.1
04 12 26 01 22 25.0 07.417 94.011 29.5 MB 6.0
04 12 26 01 25 48.0 05.438 94.172 30.0 MB 6.1
04 12 26 01 30 15.0 08.796 93.699 30.0 MB 5.5
04 12 26 01 33 22.0 07.709 93.640 25.0 MB 5.5
04 12 26 01 40 07.0 05.832 93.125 30.0 MB 5.3
04 12 26 01 48 51.0 05.389 94.443 52.5 MB 5.7
04 12 26 01 52 45.0 10.376 92.084 30.0 MB 5.2
04 12 26 01 59 14.0 08.393 92.430 30.0 MB 5.3
04 12 26 02 00 40.0 06.457 94.530 34.6 MB 6.0
04 12 26 02 15 23.0 06.262 93.504 30.0 MB 5.6
04 12 26 02 15 51.0 12.083 92.191 30.0 MB 5.3
04 12 26 02 15 59.0 12.100 92.477 28.7 MB 5.7
04 12 26 02 22 01.0 08.833 92.427 14.7 MB 5.7
04 12 26 02 30 31.0 06.685 92.988 30.0 MB 5.1
04 12 26 02 34 31.0 03.977 94.124 24.8 MB 5.7
04 12 26 02 36 08.0 12.164 92.931 25.0 MB 5.8
04 12 26 02 38 09.0 08.519 92.284 33.3 MB 5.6
```


This is a database of the first 20 earthquakes that occurred as part of the large Sumatra earthquake sequence on December 26, 2004. This file is a common form of text database where the different pieces of information are listed in different columns. The first line is the "header" which explains what each column represents: the date, time, location, and magnitude of the earthquake. Each line after represents different data points, in this case different earthquakes in the catalog. We can use this file to illustrate several ways to use `awk`.
What is the largest magnitude earthquake in this Sumatra sequence?

Answer: 9.0 ✓
Check

The correct answer is: 9
Correct
Marks for this submission: 1.00/1.00.

Question 6

Correct

1.00 points out of 1.00

Flag question

Let's first illustrate how we can match a text pattern by searching for earthquakes in our database that occurred in the first hour of December 26, 2004. If we try to do that by matching the text pattern that would represent that hour, we specify this text pattern as `/00/` in `awk`.

```
(file) jqwqecr&ltlt[your_username]> ~/groupwork/act3> awk '/00/' eqks.txt
```


Since we just specified a pattern and not an action, the output will show the completes lines that contain the text `00`.
How many lines did this `awk` command produce?

Answer: 2 ✓
Check

The correct answer is: 2
Correct
Marks for this submission: 1.00/1.00.

Question 7

Correct

1.00 points out of 1.00

Flag question

Notice that the previous `awk` command finds more than one case that has `00` on the line of text, but `00` is only in the hour column for one of those. In the first case, `00` is found in the column for hours, and in the last case, `00` is found in the column for minutes. To better select the earthquake, we could search for the text that would represent hour 00 of December 26, 2004. Which of the following commands would do that?

- Select one:
- ☐ a. `awk '00 12 26 04' eqks.txt`
 - ☐ b. `awk /12 26 04 00/ eqks.txt`
 - ☐ c. `awk /04 12 26 00/ eqks.txt`
 - ☐ d. `awk /00 04 12 26/ eqks.txt`
 - ☐ e. `awk '00 04 12 26' eqks.txt`
 - ☐ f. `awk '04 12 26 00' eqks.txt`
 - ☐ g. `awk '00 12 26 04' eqks.txt`
 - ☐ h. `awk '12 26 04 00' eqks.txt`
 - ☒ i. `awk '/04 12 26 00' eqks.txt` ✓

The correct answer is: `awk '/04 12 26 00' eqks.txt`
Correct
Marks for this submission: 1.00/1.00.

Question 8

Correct

1.00 points out of 1.00

Flag question

How many lines did this `awk` command produce?

Answer: 1 ✓
Check

The correct answer is: 1
Correct
Marks for this submission: 1.00/1.00.

Question 9

Correct

1.00 points out of 1.00

Flag question

If we just wanted to select lines with `00` in the hour column, we can use the other way to specify the pattern matching in `awk`.

```
(file) jqwqecr&ltlt[your_username]> ~/groupwork/act3> awk '94==00' eqks.txt
```



```
04 12 26 00 58 53.0 03.287 95.972 30.0 MW 9.0
```


This takes advantage of `awk`'s ability to naturally break up a line into columns that are separated by whitespace (spaces or tabs). Within `awk` we can refer to these columns by using the `$` character and then the number of the column we are interested in. In our example, we use the `==` characters to check if the number in the fourth column (`$4`) is equal to 00.
NOTE: if we are trying to match specific text instead of numbers in one of the columns, like the `MW` in the magnitude type column, we would need to use quotation marks around the text to tell `awk` to look for that exact text in the tenth column. For example:
`awk '$10=="MW"' eqks.txt`
The main comparison operators for `awk` are

Operator	Meaning
<	less than
<=	less than or equal to
==	equal to
!=	not equal to
>=	greater than or equal to
>	greater than

What command would you type on the command line to use `awk` to find the lines in `eqks.txt` that have magnitudes less than 6.0. To help guide you, use the last `awk` command above, except require the magnitude in the eleventh column to be less than 6.0.

Answer: `awk '$11<6.0' eqks.txt` ✓
Check

The correct answer is: `awk '$11<6.0' eqks.txt`
Correct
Marks for this submission: 1.00/1.00.

Question 10

Correct

1.00 points out of 1.00

Flag question

How many data lines does this command result in?
NOTE: if your command window is small, some data lines may not fit completely in your window and wrap around to use a second line of output. I only want the number of data points, so either expand your command window to full screen to make sure the lines do not wrap or pipe your output to `wc` to count the number of output lines.

Answer: 15 ✓
Check

The correct answer is: 15
Correct
Marks for this submission: 1.00/1.00.

Question 11

Correct

1.00 points out of 1.00

Flag question

Which of the following commands would select earthquakes where the magnitude is greater than or equal to 6 and send the output to **big-`eqks.txt`**.

- Select one:
- ☐ a. `awk '$1>6' > big-eqks.txt`

- ☐ b. awk '\$11>=6' >| eqs.txt
- ☐ c. awk '\$1>=6' >| eqs.txt
- ☐ d. awk '\$11>=6' big-eqs.txt
- ☐ e. awk 'MW>=6' >| big-eqs.txt
- ☒ f. awk '\$11>=6' eqs.txt >| big-eqs.txt ✓
- ☐ g. awk 'MW>=6' eqs.txt >| big-eqs.txt
- ☐ h. awk 'MW>=6' big-eqs.txt

Check

The correct answer is: awk '\$11>=6' eqs.txt >| big-eqs.txt

Correct

Marks for this submission: 1.00/1.00.

Question 12

Correct

1.00 points out of 1.00

Flag question

How many data lines are produced in the big-eqs.txt file?

Answer: 8

Check

The correct answer is: 5.5

Correct

Marks for this submission: 1.00/1.00.

Question 13

Correct

1.00 points out of 1.00

Flag question

5. awk Program Actions

The action portion of an `awk` command causes `awk` to do something when it matches a pattern. If you do not specify an action `awk` performs the default action, which is the `print` command and would be the same as writing

```
(iris) jq@clerc:~$ cat /groupwork/act3 | awk '{print}' eqs.txt
04.12.26 00.58 03.0 03.287 93.972 30.0 MB 5.0
04.12.26 01.17 10.0 04.999 94.305 30.0 MB 5.5
04.12.26 01.21 21.0 06.340 93.342 35.1 MB 6.1
... (17 more lines)
```

This action copies the line from the input file to `awk`'s standard output. You can follow a `print` command with arguments, causing `awk` to print just the arguments you specify. For example, you can print just the magnitude

```
(iris) jq@clerc:~$ cat /groupwork/act3 | awk '{print $11}' eqs.txt
```

What are the first and last numbers produced by this command?

Select one or more:

- ☐ a. 5.5
- ☐ b. 6.1
- ☒ c. 9.0 ✓ Correct, this is the first number produced by this command.
- ☒ d. 5.6 ✓ Correct, this is the last number produced by this command.
- ☐ e. 5.1

Check

The correct answer is: 9.0, 5.6

Correct

Marks for this submission: 1.00/1.00.

Question 14

Correct

1.00 points out of 1.00

Flag question

Which command would print just the earthquake depths stored in this file?

Answer: `awk '{print $9}' eqs.txt`

Check

The correct answer is: `awk '{print $9}' eqs.txt`

Correct

Marks for this submission: 1.00/1.00.

Question 15

Correct

1.00 points out of 1.00

Flag question

You can print more than one argument with the `print` command. Unless you separate items in a print command with commas, `awk` concatenates them. Commas cause `awk` to separate the items with a space by default. We can print the longitude and latitude of each earthquake with this command

```
(iris) jq@clerc:~$ cat /groupwork/act3 | awk '{print $8,$7}' eqs.txt
```

What is the last line of output produced by this command?

Select one:

- ☒ a. 92.284 08.519 ✓
- ☐ b. 03.287 95.972
- ☐ c. 08.519 92.284
- ☐ d. 95.972 03.287
- ☐ e. 08.519 09.0

Check

The correct answer is: 92.284 08.519

Correct

Marks for this submission: 1.00/1.00.

Question 16

Correct

1.00 points out of 1.00

Flag question

Additional text can also be added with the print command, which can often help to illustrate more information about the output. In this case, we can specify the directions for the longitude and latitude.

```
(iris) jq@clerc:~$ cat /groupwork/act3 | awk '{print $8,"E",$7,"N"}' eqs.txt
```

What is the last line of output produced by this command?

Answer: 92.284 E 08.519 N

Check

The correct answer is: 92.284 E 08.519 N

Correct

Marks for this submission: 1.00/1.00.

Question 17

Correct

1.00 points out of 1.00

Flag question

What does the last line of output look like if the commas are not included in the `awk` command?

Answer: 92.284E08.519N

Check

The correct answer is: 92.284E08.519N

Correct

Marks for this submission: 1.00/1.00.

Question 18

Correct

1.00 points out of 1.00

Flag question

Next you should print the longitude and latitude values where there is no space between the number and direction, but there is a space between the longitude and latitude values. Store the output in a file called `eqs-location.txt`. Which of the following commands would achieve this?

Select one:

- ☐ a. `awk '{print $8"E",$7"N"}' >| eqs-location.txt`
- ☒ b. `awk '{print $8"E",$7"N"}' eqs.txt >| eqs-location.txt` ✓
- ☐ c. `awk '{print $8"E"$7"N"}' eqs.txt >| eqs-location.txt`
- ☐ d. `awk '{print $8,"E",$7,"N"}' eqs.txt >| eqs-location.txt`
- ☐ e. `awk '{print $8,"E",$7,"N"}' >| eqs-location.txt`
- ☐ f. `awk '{print $8"E"$7"N"}' >| eqs-location.txt`

Check

The correct answer is: `awk '{print $8"E",$7"N"}' eqs.txt >| eqs-location.txt`

Correct

Marks for this submission: 1.00/1.00.

Question 19

Correct

1.00 points out of 1.00

Flag question

6. awk Arithmetic Functions and Variables

There are many other things that `awk` can do in the action part of the program, but arithmetic functions are probably the most important. We can perform mathematical adjustments to any of the input numbers. Say that you would like to reformat the dates to look like month/day/year, with the year in the full 4 digit amount (2004 instead of 04). We can do this by adding 2000 to the first column when we print out the date

```
(iris) jq@clerc:~$ cat /groupwork/act3 | awk '{print $2 "/" $3 "/" ($1+2000)}' eqs.txt
```

What is the last line of output produced by this command?

Select one:

- ☐ a. 04/12/2026
- ☒ b. 12/26/2004 ✓
- ☐ c. month/day/2000
- ☐ d. 04/12/26
- ☐ e. 12/26/04
- ☐ f. 12/26/2001

Check

The correct answer is: 12/26/2004

Correct

Marks for this submission: 1.00/1.00.

Question 20

Correct

1.00 points out of 1.00

Flag question

Notice that when the mathematical action was performed on the text in the "header" it produced a spurious result. In this case, 2000 was added the word year. Basically `awk` treats all text as zero when it performs mathematical functions, so `2000 + 0 = 2000`. We can avoid this problem by using the pattern matching aspect of `awk` to skip the first line.

The `NR` characters are a built in variable within `awk` that represent the line number (it stands for Number of the Record). In our example, we will tell `awk` to match lines where the line number is greater than 1, meaning that we want all lines after the first line.

```
(iris) jq@clerc:~$ cat /groupwork/act3 | awk 'NR>1{print $2 "/" $3 "/" ($1+2000)}' eqs.txt
```

How many lines of output are produced by this command?

Answer: 20

Check

The correct answer is: 20

Correct

Marks for this submission: 1.00/1.00.

Question 21

Correct

1.00 points out of 1.00

Flag question

We can also create our own variables in `awk`. We can achieve a similar result to our last example by setting a variable within the action part of our program. Unfortunately for you, `awk` is a little different than our shell commands, so we do not use `set` when establishing a variable, we can just specify the name and what it is equal to. In this case, we can create a variable called `year` to store the information when we calculate the four digit year. The other difference from shell variables is that we do not use the `$` character when we use variables in `awk`, we just use the name (i.e. `year`).

```
(iris) jq@clerc:~$ cat /groupwork/act3 | awk 'NR>1{year=$1+2000; print $2 "/" $3 "/" year}' eqs.txt
```

What is the first line of output produced by this command?

Select one:

- ☐ a. 04/12/26
- ☐ b. 12/26/04
- ☐ c. 12/26/2001
- ☐ d. 04/12/2026
- ☐ e. month/day/2000
- ☒ f. 12/26/2004 ✓

Check

The correct answer is: 12/26/2004

Correct

Marks for this submission: 1.00/1.00.

Question 22

Correct

1.00 points out of 1.00

Flag question

This previous question introduces the `;` character, which allows us to separate commands within the action part of an `awk` program. The first part of the action sets the `year` variable, the second part prints the information. We could add additional commands after the second part, but this should be enough of an introduction to `awk`.

Let's try another calculation, but this time with the latitude. First, which command would simply print the entire latitude column of the `eqs.txt` file?

Answer: `awk '{print $7}' eqs.txt`

Check

The correct answer is: `awk '{print $7}' eqs.txt`

Correct

Marks for this submission: 1.00/1.00.

Question 23

Correct

1.00 points out of 1.00

Flag question

Next, which of the following commands would print the distance in degrees between the earthquake latitude and the geographic North pole (90 degrees) for each event in the catalog?

Select one:

- ☐ a. `awk 'NR>1{angle=$7-90, print $angle}' eqs.txt`
- ☐ b. `awk '{angle=$7-90; print $angle}' eqs.txt`
- ☒ c. `awk 'NR>1{angle=90-$7; print angle}' eqs.txt` ✓
- ☐ d. `awk '{angle=90-$7, print angle}' eqs.txt`

Check

The correct answer is: `awk 'NR>1{angle=90-$7; print angle}' eqs.txt`

Correct

Marks for this submission: 1.00/1.00.

Question 24

Correct

1.00 points out of 1.00

Flag question

Next, which of the following commands would print the distance in radians between the earthquake latitude and the geographic North pole for each event in the catalog? Recall that there are 180 degrees per pi (3.14) radians.

Select one:

- ☒ a. `awk 'NR>1{angle=(90-$7)*3.14/180; print angle}' eqs.txt` ✓
- ☐ b. `awk 'NR>1{angle=(90-$7)*180/3.14; print angle}' eqs.txt`
- ☐ c. `awk 'NR>1{angle=$7/3.14*180; print angle}' eqs.txt`
- ☐ d. `awk 'NR>1{angle=$7/180*3.14; print angle}' eqs.txt`

Check

The correct answer is: `awk 'NR>1{angle=(90-$7)*3.14/180; print angle}' eqs.txt`

Correct

Marks for this submission: 1.00/1.00.

Question 25

Correct

1.00 points out of 1.00

Flag question

awk Final Exercise

Use `awk` to calculate the distance in kilometers of each earthquake from the equator and store the values in a file called `eqs-distance.txt`. Some helpful hints to make this calculation are that you can approximate distance on the Earth's surface using the equation for a sphere:

distance (in km) = angle (in radians) * radius (of sphere)

Notice that in the previous two question, we calculated the angle as the difference between the north pole (90) and the latitude listed in the database, but in this case we need the angle between the earthquake and the equator (0 degrees). Also we need to convert the resulting angle from degrees to radians like the previous question. You will also need the radius of the Earth, which is 6371 km.

To help you make sure your calculations are correct, the first 2 lines of the `eqs-distance.txt` file are expected to look like this:

365.312
555.582

After you complete this exercise, please paste the complete contents of your **eqks-distance.txt** file into the text box below. *For this question, do **NOT** enter the awk command into the text box, but only the complete contents of your **eqks-distance.txt** file.*

Answer: 365.312 555.582 704.618 824.315 604.371 977.575 856.767 648.16 598.926 1153.17 932.786 762.077 695.95 1342.89 1369.23 981.687 742.961 441.998 1351.89 946.789

Check

The correct answer is: 365.312 555.582 704.618 824.315 604.371 977.575 856.767 648.16 598.926 1153.17 932.786 762.077 695.95 1342.89 1369.23 981.687 742.961 441.998 1351.89 946.789

Correct

Marks for this submission: 1.00/1.00.

[Finish review](#)

You are logged in as **Dilshad Raza** ([Log out](#))

IRIS2022SSBW