**PSG COLLEGE OF TECHNOLOGY**

**UNIX SHELL AND SYSTEM PROGRAMMING LAB**

**Awk Problem Sheet Answers**

**Some string functions in awk**

index(string,search)  
length(string)  
split(string,array,separator)  
substr(string,position)  
substr(string,position,max)  
tolower(string)  
toupper(string)

1. Write an awk command to print the lines and line numberin the given input file .

$ cat>cmds.awk{print NR, $0 }$ awk -f cmds.awkfileinput

2. Write an awk command to print first field and second field only if third field value is >=50 in the given input file. (input field separator is “:” and output field separator is “,”)

**input data file**

$cat > file1sachin:10:100rahul:11:35rohit:12:89

$awk -F':' '$3>=50 {print $1”,”$2}' file1

**output:**

sachin,10rohit,12

3 . Consider the marks.txt is a file that contains one record per line( comma separate fields) of the student data in the form of studentid, student name, Telugu marks, English marks, Maths Marks, Science marks, Social Marks. Write an awk script to generate result for every students in the form of studentid, studentname, Total Marks and result. Result is PASS if marks is >=30 in TELUGU and English, and if marks>=40 in other subjects. Result is fail otherwise.

**Input file**

$cat > marks.txt

1001,name1,99,69,85,56,75

1002,name2,89,69,65,56,55

1003,name3,50,50,50,55,55

1004,name4,69,29,85,56,75

1005,name5,99,69,85,56,11

createmarks.awk script file

$cat >marks.awk

{

total=$3+$4+$5+$6+$7

if($3>=30 && $4>=30 && $5>=40 && $6>=40 && $7>=40)

print $1,$2,total, "Pass";

else

print $1,$2,total, "fail";

}

$awk -F “,” -f marks.awk marks.txt

4 Write an awk program to print the fields 1 and 4 of a file that is passed as command line argument. The file contains lines of information that is separated by “,” as delimiter. The awk program must print at the end the average of all 4th field data.

**Input file**

$cat > data

12,13,14,15,one

22,23,24,25,two

34,23,45,23,three

44,55,66,77,four

$awk -F',' '{print $1,$2,$3,$4,($1+$2+$3+$4)/4}' data

5.Write an awk program to demonstrate user defined functions and system command.

$cat > data

12,13,14,15,one

22,23,24,25,two

34,23,45,23,three

44,55,66,77,four

$cat >user.awk

{

if($3>0)

display($3)

}

function display(name)

{

print name

}

$awk -F',' -f user.awk data

1. Write an awk script to count the number of lines in a file that do not contain vowels.

**Input file**

$cat > input

this is one

213

BCDEFG

This is last line

$cat vowels.awk

BEGIN{count=0}

!/[aeiou]/ {count++;print}

END{print "Number of lines="count}

$awk -f vowels.awk input

1. Write an awk script to find the number of characters, words and lines in a file.

**Input file:**

$cat > file7

This is a file

YEs NO

1234

$cat >lines.awk

BEGIN{words=0;characters=0}

{

character+=length($0);

words+=NF;

}

END{print "lines=",NR," words=",words," Characters=",character}

$awk -f lines.awk file7

1. Awk command to read a list of integers from the file and print the total

awk '{ sum += $1 }; END { print sum }' file

1. Display the number of columns in each row.

awk '{print NF}' input\_file

1. Display the line numbers from 1.

awk '{print NR}' input\_file

1. Display the total number of lines in the filE

awk 'END {print NR}' input\_file

1. awk script to insert a new line ‘9Z’ after every 2 line in file.txt

The input "file.txt" contains the below data:

1 A

2 B

3 C

4 D

5 E

6 F

The required output data after inserting a new line looks as

1 A

2 B

9 Z

3 C

4 D

9 Z

5 E

6 F

9 Z

awk '{

if(NR%2 == 0)

{

print $0"\n9 Z";

}

else

{

print $0

}

}' file.txt

1. The input file contains the data.

AAA 1

BBB 2

CCC 3

AAA 4

AAA 5

BBB 6

CCC 7

AAA 8

BBB 9

AAA 0

Replace the fourth occurrence of the first field "AAA" with "ZZZ" in the file.  
  
The required output is:

AAA 1

BBB 2

CCC 3

AAA 4

AAA 5

BBB 6

CCC 7

ZZZ 8

BBB 9

AAA 0

The awk command for getting this output is

awk 'BEGIN {count=0}

{

if($1 == "AAA")

{

count++

}

if(count == 4)

{

sub("AAA","ZZZ",$1)

}

}

{

print $0

}' file.txt

15. The input file data:

A 10

B 39

C 22

D 44

E 75

F 89

G 67

Write an awk script to get the second field and then find the sum the even and odd lines.  
The required output is

174, 172

The awk command for producing this output is

awk '{

if(NR%2 == 1)

{

sum\_e = sum\_e + $2

}

else

{

sum\_o = sum\_o + $2

}

}

END { printsum\_e,sum\_o }' file.txt

16. Fibonacci series using awk command  
  
wk ' BEGIN{

for(i=0;i<=10;i++)

{

if (i<=1 )

{

x=0;

y=1;

printi;

}

else

{

z=x+y;

print z;

x=y;

y=z;

}

}

}'

17. Remove leading zeros from a file using the awk command. The input file contains the below data.

0012345

05678

01010

00001

After removing the leading zeros, the output should contain the below data.

12345

5678

1010

1

The awk command for this is.

awk '{print $1 + 0}' file.txt

awk '{printf "%d\n",$0}' file.txt

18. The file employee.txt contains the following data

Tom Manager Sales $5,000

Jason Developer Technology $5,500

Sanjay Sysadmin Technology $7,000

Nisha Manager Marketing $9,500

Randy DBA Technology $6,000

* 1. write an awk script to produce the following output

Name Designation Department Salary

Tom Manager Sales $5,000

Jason Developer Technology $5,500

Sanjay Sysadmin Technology $7,000

Nisha Manager Marketing $9,500

Randy DBA Technology $6,000

Report Generated

--------------

$ awk 'BEGIN {print "Name\tDesignation\tDepartment\tSalary";}

> {print $1,"\t",$2,"\t",$3,"\t",$NF;}

>END{print "Report Generated\n--------------";

> }' employee.txt

* 1. Print the list of employees in Technology department

$ awk '$4 ~/Technology/' employee.txt

200 Jason Developer Technology $5,500

300 SanjaySysadmin Technology $7,000

500 Randy DBA Technology $6,000

* 1. Print number of employees in Technology department

$ awk 'BEGIN { count=0;}

$4 ~ /Technology/ { count++; }

END { print "Number of employees in Technology Dept =",count;}' employee.txt

1. To print non-empty line from a file.

$ awk 'NF > 0' sample.txt

1. To print the length of the longest input line.

$ awk '{ if (length($0) > max) max = length($0) } END { print max }' sample.txt

1. To print seven random numbers from zero to 100, inclusive.

$ awk 'BEGIN { for (i = 1; i<= 7; i++) print int(101 \* rand()) }'

1. To count the lines in a file

$ awk 'END { print NR }' sample.txt

1. Awk script that asks for a number and then squares it.

BEGIN {

print "type a number";

}

{

print "The square of ", $1, " is ", $1\*$1;

print "type another number";

}

END {

print "Done"

}