UNIX ASSIGNMENT – 2

NAME:C.PAVITHRA

ROLL NO:422127

SECTION: A

Write shell script for pattern matching using

1. AWK:

Code :

```
echo 'print all the rows and columns in file1:'

awk '{print $0}' file1.txt

echo ''

echo 'print 1 and 4 columns of file1:'

awk '{print $1,$4}' file1.txt

echo''

echo 'print sum of numbers in column 3 of file1:'

awk '{sum+=$3} END {print sum}' file1.txt

echo''
echo 'print avg of column 3 values of file1:'

awk '{sum+=$3} END {print sum/NR}' file1.txt

echo''

echo 'print 2 and 3 characters of column 1 of file1:'

awk '{print substr($1,2,3) }' file1.txt

echo''

echo 'print last column of file1:'

awk '{print $NF }' file1.txt

echo

echo 'print junior for salary less than 15000 otherwise print senior:'
```

```
awk '{if ($3 > 15000) {print "senior" } else {print "junior"}}' file1.txt

echo''

echo 'print the square of the values of column 4 of file1:'

awk 'function square(x) {return x*x } {print square($4) }' file1.txt
```

output :

```
print all the rows and columns in file1:
s01 em1 10000 25
s02 em2 12000 24
s03 em3 11020 22
s04 em4 13000 23
s05 em5 19000 26
s06 em6 20000 25
s07 em7 25000 27
s08 em8 32000 28

print 1 and 4 columns of file1:
s01 25
s02 24
s03 22
s04 23
s05 26
s06 25
s07 27
s08 28

print sum of numbers in column 3 of file1:
142020

print avg of column 3 values of file1:
17752.5

print 2 and 3 characters of column 1 of file1:
01
02
03
04
05
06
07
08

print last column of file1:
25
24
```

```
22
23
26
25
27
28

print junior for salary less than 15000 otherwise print senior:
junior
junior
junior
junior
senior
senior
senior
senior

print the square of the values of column 4 of file1:
625
576
484
529
676
625
729
784
```

2. SED:

CODE :

```
'print file1:'
cat file1.txt
echo
echo 'replace s01 with 01 line by line:'
sed 's/s01/01/' file1.txt
echo
echo 'after above command the original file doesn’t change print file1:'
cat file1.txt
echo
echo'replace 25 in the place of 21:'
sed 's/25/21/g' file1.txt
echo
echo'to delete a line no. 2:'
sed '2d' file1.txt
echo
echo'to delete a last line:'
sed '$d' file1.txt
echo
echo'to delete lines from 1 to 4:'
sed '1,4d' file1.txt
```

```
echo
echo'to delete lines from 4 to last:'
sed '4,$d' file1.txt
echo
echo 'to delete pattern matching line:'
sed '/22/d' file1.txt
echo
```

OUTPUT:

```
print file1:
s01 em1 10000 25
s02 em2 12000 24
s03 em3 11020 22
s04 em4 13000 23
s05 em5 19000 26
s06 em6 20000 25
s07 em7 25000 27
s08 em8 32000 28

replace s01 with 01 line by line:
01 em1 10000 25
s02 em2 12000 24
s03 em3 11020 22
s04 em4 13000 23
s05 em5 19000 26
s06 em6 20000 25
s07 em7 25000 27
s08 em8 32000 28

after above command the original file doesn’t change print file1:
s01 em1 10000 25
s02 em2 12000 24
s03 em3 11020 22
s04 em4 13000 23
s05 em5 19000 26
s06 em6 20000 25
s07 em7 25000 27
s08 em8 32000 28

replace 25 in the place of 21:
s01 em1 10000 21
s02 em2 12000 24
s03 em3 11020 22
s04 em4 13000 23
s05 em5 19000 26
s06 em6 20000 21
s07 em7 21000 27
s08 em8 32000 28
```

```
to delete a line no. 2:
s01 em1 10000 25
s03 em3 11020 22
s04 em4 13000 23
s05 em5 19000 26
s06 em6 20000 25
s07 em7 25000 27
s08 em8 32000 28

to delete a last line:
s01 em1 10000 25
s02 em2 12000 24
s03 em3 11020 22
s04 em4 13000 23
s05 em5 19000 26
s06 em6 20000 25
s07 em7 25000 27

to delete lines from 1 to 4:
s05 em5 19000 26
s06 em6 20000 25
s07 em7 25000 27
s08 em8 32000 28

to delete lines from 4 to last:
s01 em1 10000 25
s02 em2 12000 24
s03 em3 11020 22

to delete pattern matching line:
s01 em1 10000 25
s02 em2 12000 24
s04 em4 13000 23
s05 em5 19000 26
s06 em6 20000 25
s07 em7 25000 27
s08 em8 32000 28
```

3. TAR:

CODE :

```
echo'archive of every .txt file :'
tar cvf file.tar *.txt
echo
echo 'extract files through archives:'
tar xvf file.tar
echo
echo'makes a tar file known as file.tar.gz :'
```

```
tar cvzf file.tar.gz *.txt
echo
echo'extract the files through file.tar.gz tar archived files :'
tar xvzf file.tar.gz
echo
echo'to create and compress archive files :'
tar cvfj file.tar.tbz all.txt
echo
echo'to untar or extract more than one file from tar.bz2, tar.gz, and a
tar archive file :'
tar -jxvf file.tar.tbz "all.txt"
echo
echo' to show the archive file's size in kilobytes  :'
tar -czf - file.tar | wc -c
echo
echo'for updating an existing tar file  :'
tar rvf file.tar *.txt
echo
echo'to list the whole archive file's list :'
tar tf file.tar
echo
echo'to view the archieve:'
tar -tvf file.tar
echo
echo'to pass the file name as argument:'
tar tvf file.tar all.txt
echo
```

OUTPUT:

```
archive of every .txt file:
all.txt
awk_output.txt
file1.txt
hello.txt
sed_output.txt

extract files through archives:
all.txt
awk_output.txt
file1.txt
hello.txt
sed_output.txt

makes a tar file known as file.tar.gz:
all.txt
awk_output.txt
file1.txt
hello.txt
sed_output.txt
```

extract the files through file.tar.gz tar archived files:

```
all.txt
awk_output.txt
file1.txt
hello.txt
sed_output.txt
```

to create and compress archive files:

```
all.txt
```

to **untar** or extract more than one file from tar.bz2, tar.gz, and a tar archive file:

```
all.txt
```

to show the archive file's size in kilobytes :

```
625
```

for updating an existing tar file :

```
all.txt
awk_output.txt
file1.txt
hello.txt
sed_output.txt
```

to list the whole archive file's list:

```
all.txt
awk_output.txt
file1.txt
hello.txt
sed_output.txt
all.txt
awk_output.txt
file1.txt
hello.txt
sed_output.txt
```

to view the archieve:

```
-rw-rw-r-- student/student   0 2024-02-28 15:52 all.txt
-rw-rw-r-- student/student 697 2024-02-28 15:10 awk_output.txt
-rw-rw-r-- student/student 136 2024-02-28 15:00 file1.txt
-rw-rw-r-- student/student   0 2024-02-28 16:03 hello.txt
-rw-rw-r-- student/student 1049 2024-02-28 15:26 sed_output.txt
-rw-rw-r-- student/student    0 2024-02-28 15:52 all.txt
-rw-rw-r-- student/student  697 2024-02-28 15:10 awk_output.txt
-rw-rw-r-- student/student  136 2024-02-28 15:00 file1.txt
-rw-rw-r-- student/student    0 2024-02-28 16:03 hello.txt
-rw-rw-r-- student/student 1049 2024-02-28 15:26 sed_output.txt
```

to pass the file name as the argument:

```
-rw-rw-r-- student/student   0 2024-02-28 15:52 all.txt
```

```
-rw-rw-r-- student/student   0 2024-02-28 15:52 all.txt
```

4. CPIO:

CODE :

```
find file1.txt file2.txt newfile | cpio -o > archive.cpio
echo -e "\n"
cpio -i < archive.cpio
echo -e "\n"
cpio -t < archive.cpio
echo -e "\n"
find file1.txt file2.txt newfile | cpio -o -H odc > archive.cpio
echo -e "\n"
cpio -o -F archive_file1.cpio < file1.txt
echo -e "\n"
# Create a CPIO archive with verbose output
find file1.txt file2.txt newfile | cpio -vo > archive.cpio
echo -e "\n"
cpio -i -d < archive.cpio
echo -e "\n"
cpio -i -m < archive.cpio
echo -e "\n"
```

OUTPUT:

```
1 block


cpio: file1.txt not created: newer or same age version exists
cpio: file2.txt not created: newer or same age version exists
cpio: newfile not created: newer or same age version exists
1 block


file1.txt
file2.txt
newfile
1 block


1 block


cpio: s01 em1 10000 25: Cannot stat: No such file or directory
cpio: s02 em2 12000 24: Cannot stat: No such file or directory
cpio: s03 em3 11020 22: Cannot stat: No such file or directory
cpio: s04 em4 13000 23: Cannot stat: No such file or directory
cpio: s05 em5 19000 26: Cannot stat: No such file or directory
cpio: s06 em6 20000 25: Cannot stat: No such file or directory
cpio: s07 em7 25000 27: Cannot stat: No such file or directory
cpio: s08 em8 32000 28: Cannot stat: No such file or directory
1 block
```

```
file1.txt
file2.txt
newfile
1 block
```

```
cpio: file1.txt not created: newer or same age version exists
cpio: file2.txt not created: newer or same age version exists
cpio: newfile not created: newer or same age version exists
1 block
```

```
cpio: file1.txt not created: newer or same age version exists
cpio: file2.txt not created: newer or same age version exists
cpio: newfile not created: newer or same age version exists
1 block
```