EXERCISE 1:

1. **Launch a terminal.**

2. **Create and save the following file. Note that the fields are separated by one or more spaces (randomly). The first field is the first name, the second field is the last name, and the third field is the age.**

   John Adams 55
   George Bull 77
   Anne Blue 99
   Janet Blue 67
   Ben Benjamin 78
   Ted White 32

Answer:

```
dikshu:~$cat>file1.txt
John Adams 55
George Bull 77
Anne Blue 99
Janet Blue 67
Ben Benjamin 78
Ted White 32
```

3. **Use a one-line sed command to reorganize the file using the comma/tab pattern shown in the following file. Note that the last name is before the first name, and there is only one space between the names and a space before the numbers. Use the same name for the new file.**

   Adams, John 55
   Bull, George 77
   Blue, Anne 99
   Blue, Janet 67
   Benjamin, Ben 78
   White, Ted 32

Answer:

```
dikshu:~$sed -E -i.bak "1,6 s/^([^\s]*)\s+([^\s]*[s]*)\s+([0-
9]*)\s*$/\2, \1 \3/g" file1.txt

dikshu:~$cat file1.txt
Adams, John 55
Bull, George 77
Blue, Anne 99
Blue, Janet 67
Benjamin, Ben 78
White, Ted 32
```

4. **Sort the file first according to the last name and then according to the age. Watch out for the comma after the last name. Use the same name for the new file.**

Answer:

```
dikshu:~$sort +0 -1 +2 -3n file1.txt
Adams, John 55
Benjamin, Ben 78
Blue, Janet 67
Blue, Anne 99
Bull, George 77
White, Ted 32
```

5. **Use a sed script and a sed command to put a set of five asterisks at the beginning and end of each line that contains the pattern Blue.**

Answer:

```
dikshu:~$cat>script1.sed
s/.*Blue.*$/*****&*****/

dikshu:~$sed -f script1.sed file1.txt
Adams, John 55
Bull, George 77
*****Blue, Anne 99*****
*****Blue, Janet 67*****
Benjamin, Ben 78
White, Ted 32
```

6. **Use the cat command to insert a line number at the beginning of each line in a file.**

Answer:

```
dikshu:~$cat -n file1.txt
     1  Adams, John 55
     2  Bull, George 77
     3  Blue, Anne 99
     4  Blue, Janet 67
     5  Benjamin, Ben 78
     6  White, Ted 32
```

7. **Write a sed script and a sed command to split the file into three files. The first file, called f1, contains lines 2 and 3. The second file, called f2, contains lines 4 and 5. The third file, called f3, contains lines 1 and 6.**

Answer:

```
dikshu:~$cat>script1.sed
2w f1
3w f1
4w f2
5w f2
1w f3
6w f3

dikshu:~$sed -n -f script1.sed file1.txt
```

8. **Print all the files created in this session and verify the output.**

Answer:

```
dikshu:~$cat f1
Bull, George 77
Blue, Anne 99

dikshu:~$cat f2
Blue, Janet 67
Benjamin, Ben 78

dikshu:~$cat f3
Adams, John 55
White, Ted 32
```

9. **Quit the terminal.**

```
dikshu:~$exit
```

EXERCISE 2:

1. **Launch a terminal.**

2. **Create the following file and call it a7-e4-f1. Each line in the file is an absolute pathname of a file.**

```
bin/date
bin/programs/cal
usr/bin/date usr/report/file1
usr/report/1etters/lett1
/spool/mails
```

Answer:

```
dikshu:~$cat>a7-e4-f1
```

```
bin/date
bin/programs/cal
usr/bin/date
usr/report/file1
usr/report/1etters/lett1
/spool/mails
```

3. **Write a sed script (a7-e4-f2) and a sed command to extract the lowest level direc- tory and the name of the file from the path (separated by spaces) and store it in a file called a7-e4-f3. The file should look like the following (directory then file}:**

   /bin date
   /bin/programs cal
   /usr/bin date
   /usr/report file1
   /usr/report/letters lett1
   /spool mails

Answer:

dikshu:~$cat a7-e4-f2.sed

```
s/\/\([A-Za-z0-9]*\)$/ \1/
s/^[^\/]/\/&/
```

dikshu:~$sed -f a7-e4-f2.sed a7-e4-f1>a7-e4-f3

dikshu:~$cat a7-e4-f3

```
/bin date
/bin/programs cal
/usr/bin date
/usr/report file1
/usr/report/1etters lett1
/spool mails
```

4. **Quit the terminal.**

   dikshu:~$exit

EXERCISE 3:

1. **Launch a terminal.**

2. **Create the following file and call it a7-e6-f1. The file is a C program that multiplies two numbers. It contains some comments which begin with the two-character token /* and end with the two-character token */. In this program, comments can be on one line or can span more than a line**

```
/* This program reads two integer numbers from the keyboard and
prints their product.
Written by:
Date:

*/

/* Statements */

scanf ("%d", &number1); scanf ("%d", &number2); result = number1 *
number2; printf ("%d", result); return 0;
} /* main */
```

Answer:

```
dikshu:~$cat>a7-e6-f2
/* This program reads two integer numbers from the keyboard and
prints their product.
Written by:
Date:

*/

/* Statements */

scanf ("%d", &number1);
scanf ("%d", &number2);
result = number1 * number2;
printf ("%d", result); return 0;
} /* main */
```

3. **Write a sed script a7-e6-f2.sed and a sed command to delete the comments from the file. Call the new file a7-e6-f3.**

Answer:

```
dikshu:~$cat>a7-e6-f2
/\/\*.*\*\//d
/\/\*/,/\*\//d

dikshu:~$sed -f a7-e6-f2 a7-e6-f1


scanf ("%d", &number1);
scanf ("%d", &number2);
result = number1 * number2;
printf ("%d", result);
return 0;
```

4. **Quit the terminal.**

```
dikshu:~$exit
```

EXERCISE 4:

1. **Create the file named workers.txt with the following**

First Name	Last Name	Rate	Hours
George	White	18.00	23
Mark	Red	18.10	20
Mary	Blue	10.89	25
Dan	Black	12.00	0
Susan	Green	18.00	40
Nora	Brown	17.20	46
Bruce	Purple	12.20	52
John	Gray	11.00	39
Bob	Gold	15.00	45
Steve	Silver	14.67	25

Answer:

```
dikshu:~$cat>workers.txt
George    White    18.00    23
Mark    Red    18.10    20
Mary    Blue    10.89    25
Dan    Black    12.00    0
Susan    Green    18.00    40
Nora    Brown    17.20    46
Bruce    Purple    12.20    52
John    Gray    11.00    39
Bob    Gold    15.00    45
Steve    Silver    14.67    25
```

a. Write awk command to print the first and last name who did not work in the last week

Answer:

```
dikshu:~$awk '$4==0 {print $1" " $2}' workers.txt
Dan Black
```

b. Write awk command to print the record of the employee whose rate is $15 or more

Answer:

```
dikshu:~$awk '$3>=15 {print}' workers.txt
George    White    18.00    23
Mark    Red    18.10    20
Susan    Green    18.00    40
Nora    Brown    17.20    46
Bob    Gold    15.00    45
```

c. Write awk command to print the record whose first name is Mary

Answer:

```
dikshu:~$awk '$1=="Mary" {print}' workers.txt
Mary    Blue    10.89    25
```

d. Write awk command to print the record of the employee whose rate between $1 and $18

Answer:

```
dikshu:~$awk '$3>=1 && $3<=18 {print}' workers.txt
George  White    18.00    23
Mary    Blue    10.89    25
Dan    Black    12.00    0
Susan    Green    18.00    40
Nora    Brown    17.20    46
Bruce    Purple    12.20    52
John    Gray    11.00    39
Bob    Gold    15.00    45
Steve    Silver    14.67    25
```

2. Create a file sales.txt with the following contents

Month	**Sales**
January	**20**
February	**30**
March	**43**
February	**34**
January	**12**
June	**89**
May	**97**
June	**60**
July	**23**
August	**13**
August	**45**
October	**56**
October	**45**
November	**34**

Answer:

```
dikshu:~$cat>sales.txt
January    20
February    30
March    43
February    34
January    12
June    89
May    97
June    60
July    23
```

```
August    13
August    45
October    56
October    45
November    34
```

a. Write awk command to find total sales

Answer:

dikshu:~$cat>totalsales.awk

```
BEGIN{total=0}
{total+=$2}
END{print "TOTAL SALES = ",total}
```

dikshu:~$awk -f totalsales.awk sales.txt

```
TOTAL SALES =  601
```

b. Write awk script to find the total sales in every month

Answer:

dikshu:~$cat monthly.awk

```
BEGIN{
     jan=0
     feb=0
     mar=0
     apr=0
     may=0
     jun=0
     july=0
     aug=0
     sep=0
     oct=0
     nov=0
     dec=0
}
{
if($1=="January"){
jan+=$2
}
else if($1=="February"){
feb+=$2
}
else if($1=="March"){
mar+=$2
}
else if($1=="April"){
apr+=$2
}
```

```
else if($1=="May"){
may+=$2
}
else if($1=="June"){
jun+=$2
}
else if($1=="July"){
july+=$2
}
else if($1=="August"){
aug+=$2
}
else if($1=="September"){
sep+=$2
}
else if($1=="November"){
nov+=$2
}
else if($1=="October"){
oct+=$2
}
else if($1=="December"){
dec+=$2
}
}
END{
     print "January: " jan
     print "February: " feb
     print "March: " mar
     print "April: " apr
     print "May: " may
     print "June: " jun
     print "July: " july
     print "August: " aug
     print "September: " sep
     print "October: " oct
     print "November: " nov
     print "December: "dec
}
```

dikshu:~$awk -f monthly.awk sales.txt

```
January:  32
February: 64
March: 43
April: 0
May: 97
June: 149
July: 23
August: 58
September: 0
October: 101
November: 34
```

```
December: 0
```

c. Write awk script to find the months with no sales

Answer:

dikshu:~$awk -f monthly.awk sales.txt>monthlysales.txt

dikshu:~$cat monthlysales.txt

```
January:  32
February: 64
March: 43
April: 0
May: 97
June: 149
July: 23
August: 58
September: 0
October: 101
November: 34
December: 0
```

dikshu:~$cat>nosales.awk

```
BEGIN{FS=": "}
{
     if($2==0)
          {print $1}
}
^C
```

dikshu:~$awk -f nosales.awk monthlysales.txt

```
April
September
December
```