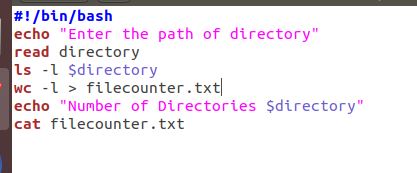
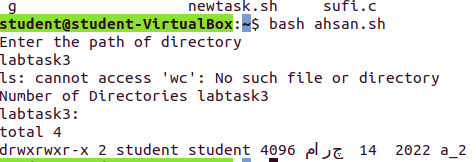
**Task No. 1:** Write a shell script to count all files and folders present in directory and stored the output into a text file and display its content on the terminal.

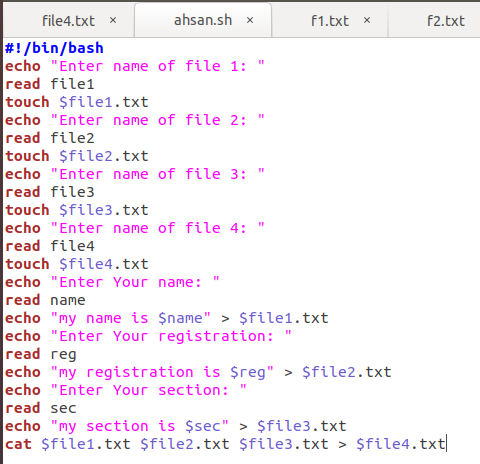
**Solution:**

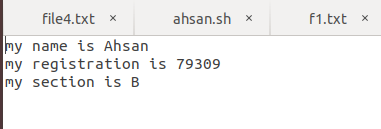


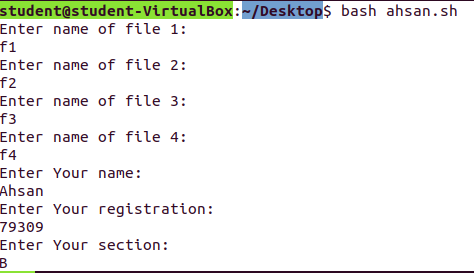
**Output:**



**Task No. 2:** Write a single shell script that creates four different files, while taking the names of all created files as input from the user. As the files content, insert your name in the first file, registration number in the second and section details in the third. These should be followed by merging the contents of all three files into the fourth one.

**Solution:**





**Task No. 3:** Write a shell script that either performs a file sort, file search or directory listing operation based on the user’s selection of the operation he/she would like to execute

Text, letter

Description automatically generated**Solution:**

#!bin/bash/

cd Desktop

echo "Please Choose From Below "

echo " 1 ) File Sort"

echo " 2 ) File Search"

echo " 3 ) Directory Listing"

read number

if [ $number == 1 ];

then

echo "---------------------------------"

echo "-------------File Sort-----------"

echo "---------------------------------"

echo "Enter file Name to Sort "

read filename

sort $filename.txt

elif [ $number == 2 ];

then

echo "---------------------------------"

echo "---------File Search ------------"

echo "---------------------------------"

echo "Enter File Name To Search"

read filenametosearch

find . -iname "$filenametosearch.sh"

elif [ $number == 3 ];

then

echo "---------------------------------"

echo "---------Directory Listing-------"

echo "---------------------------------"

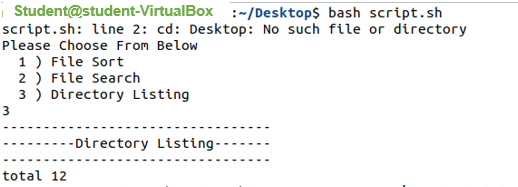
ls -l

else

echo "Please Enter Correct Option"

fi

**Output:**



**Task No. 4:** Write a C program that takes values of two matrices of size (𝑚×1) and (1×𝑛) as input from the user. Multiply the above two matrixes and store the resulting (𝑚×𝑛) matrix in a 2D array. Display the contents of the first and second matrices and also the resulting matrix. Achieve alignment in the displayed content as much possible.

**Solution:**

#include <stdio.h>

int main() {

int m, n, i, j, k;

printf("Enter the value of m: ");

scanf("%d", &m);

printf("Enter the value of n: ");

scanf("%d", &n);

int matrix1[m][1], matrix2[1][n], result[m][n];

printf("Enter the values of the first matrix:\n");

for (i = 0; i < m; i++) {

for (j = 0; j < n; j++) {

printf("Enter Value [%d][%d] : ",i,j);

scanf("%d", &matrix1[i][j]);}}

printf("Enter the values of the second matrix:\n");

for (i = 0; i < m; i++) {

for (j = 0; j < n; j++) {

printf("Enter Value [%d][%d] : ",i,j);

scanf("%d", &matrix2[i][j]);}}

for (i = 0; i < m; i++) {

for (j = 0; j < n; j++) {

result[i][j] = 0;

for (k = 0; k < 1; k++) {

result[i][j] += matrix1[i][k] \* matrix2[k][j];}}}

printf("First Matrix:\n");

for (i = 0; i < m; i++) {

for (j = 0; j < n; j++) {

printf("%d ", matrix1[i][j]);

}

printf("\n");}

printf("Second Matrix:\n");

for (i = 0; i < m; i++) {

for (j = 0; j < n; j++) {

printf("%d ", matrix2[i][j]);}

printf("\n");}

printf("Result Matrix:\n");

for (i = 0; i < m; i++) {

for (j = 0; j < n; j++) {

printf("%d ", result[i][j]);}

printf("\n");} return 0;}

**Output:**

