Faculty of Engineering & Technology

Electrical &Computer Engineering Department

Linux Lab / ENCS3130

Report Project 1

\

Abstract

This report shows the results and discuss some cases and code lines for the first project in linux laboratory, shell scripting project.

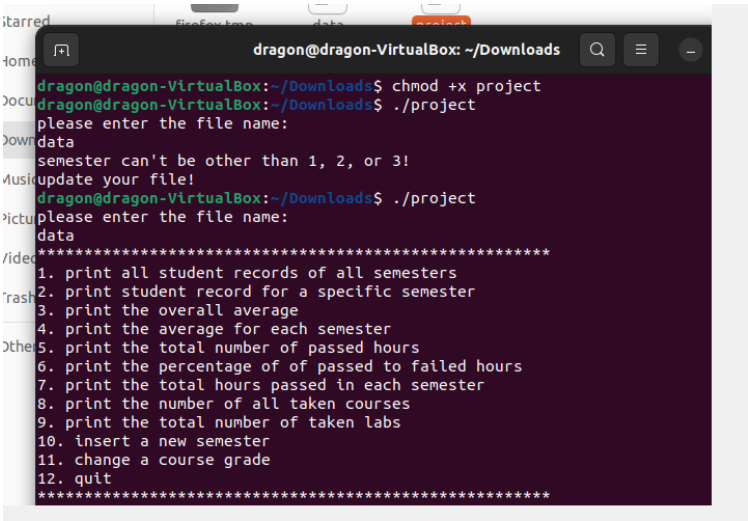
Contents

The main menu

This image show the main menu for the program, which will keep showing after each task, until the user enters 12, it will stop the running.

Before showing the main menu, the user will be asked to enter the file name, then it will be checked if it exists.If line exists it will be read line by line and split upon the format given before for years, semesters etc.

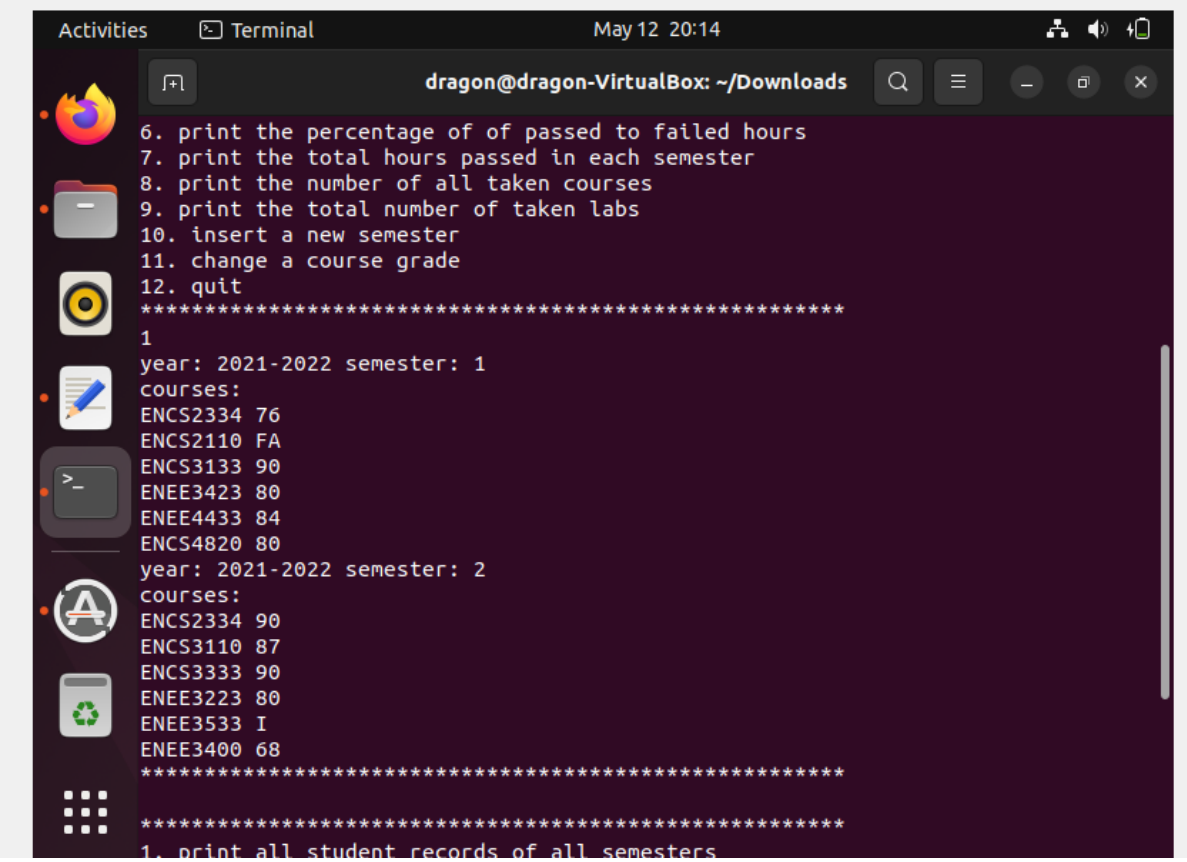
Then the main menu will appear to make actions on data.



Option 1: print all student records of all semesters:

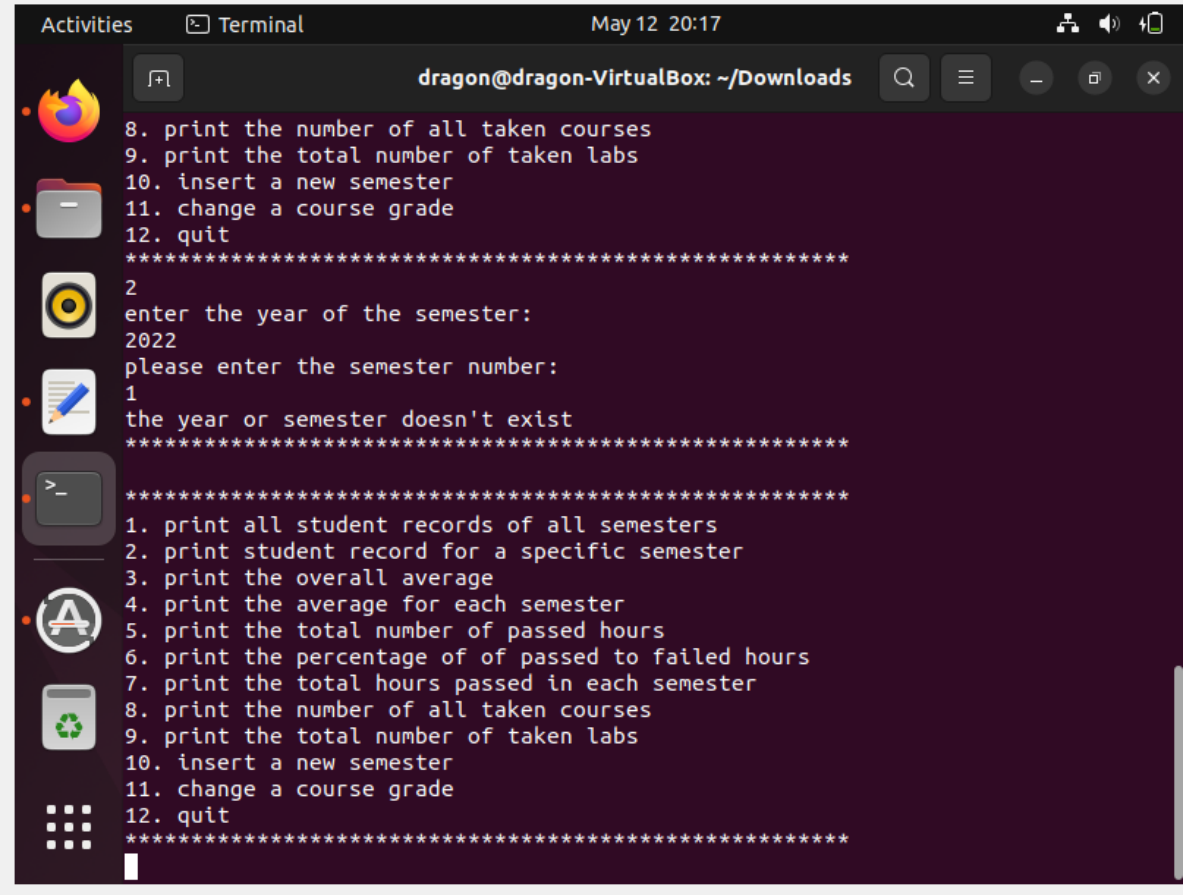
This will ask the user to enter the year in this format (YYYY-YYYY) then the semester number.

For semester number to be valid it must have a value between 1 and 3


Activities
Terminal
May 12 20:14
dragon@dragon-VirtualBox: ~/Downloads
6. print the percentage of of passed to failed hours
7. print the total hours passed in each semester
8. print the number of all taken courses
9. print the total number of taken labs
10. insert a new semester
11. change a course grade
12. quit
****
1
year: 2021-2022 semester: 1
courses:
ENCS2334 76
ENCS2110 FA
ENCS3133 90
ENEE3423 80
ENEE4433 84
ENCS4820 80
year: 2021-2022 semester: 2
courses:
ENCS2334 90
ENCS3110 87
ENCS3333 90
ENEE3223 80
ENEE3533 I
ENEE3400 68
****
****
1. print all student records of all semesters

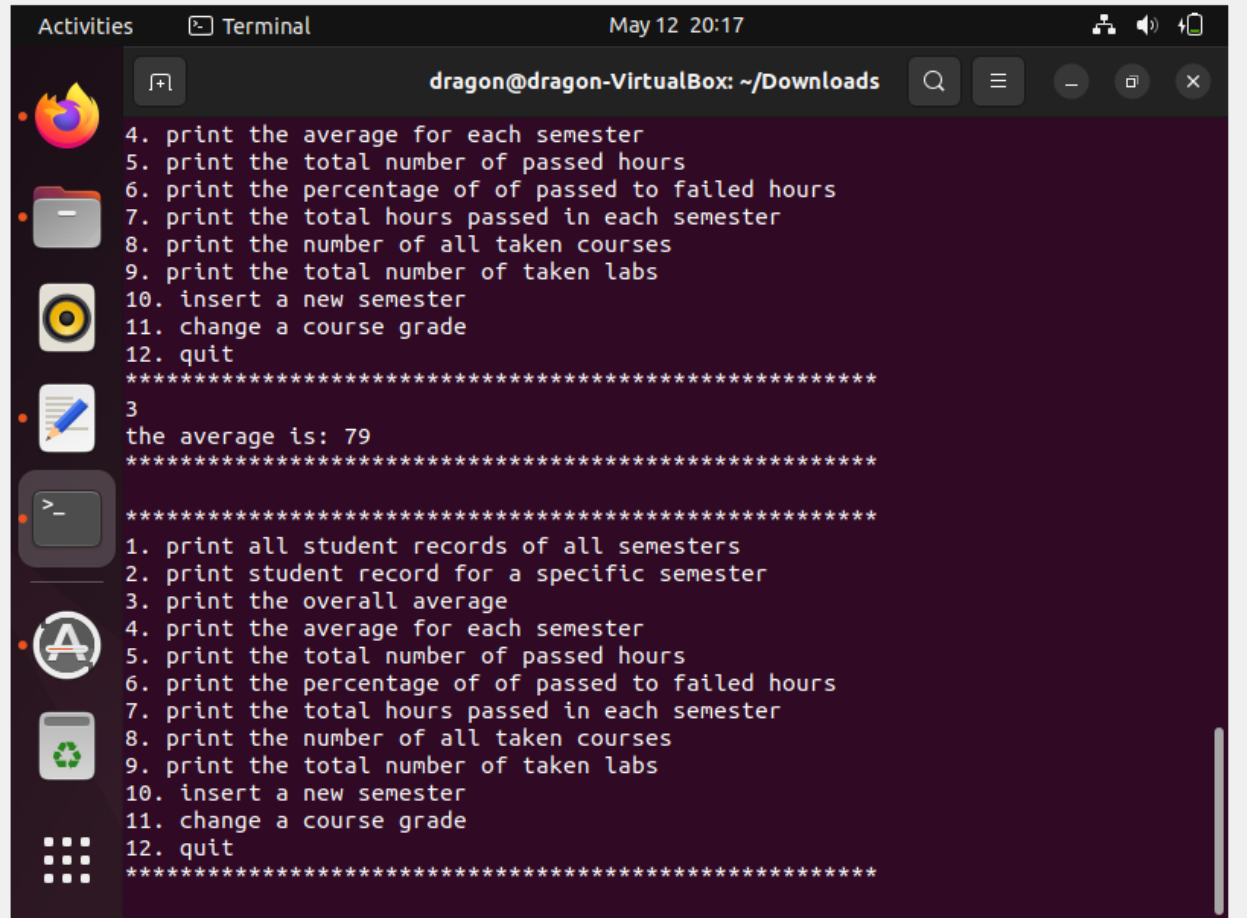
Option 2: print student records for specific semester

This ask the user to enter the year and the semester if the year of the semester does not exist it will show a message telling the user that one of them does not exist and return to show the main menu.



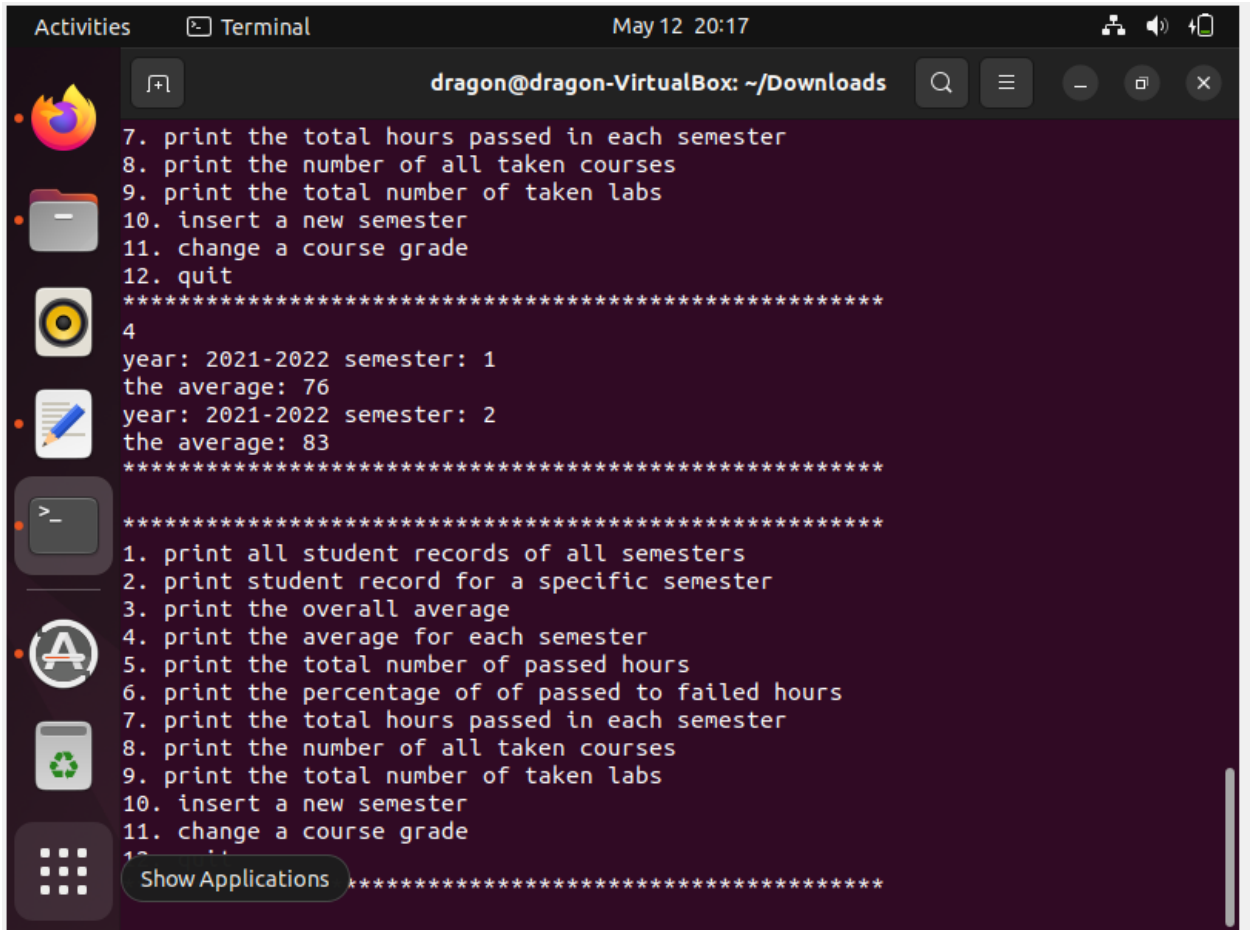
Option 3: calculate the average

This will calculate the average for all marks, taking in consideration all options, FA wii be counted as 50, F as 55.



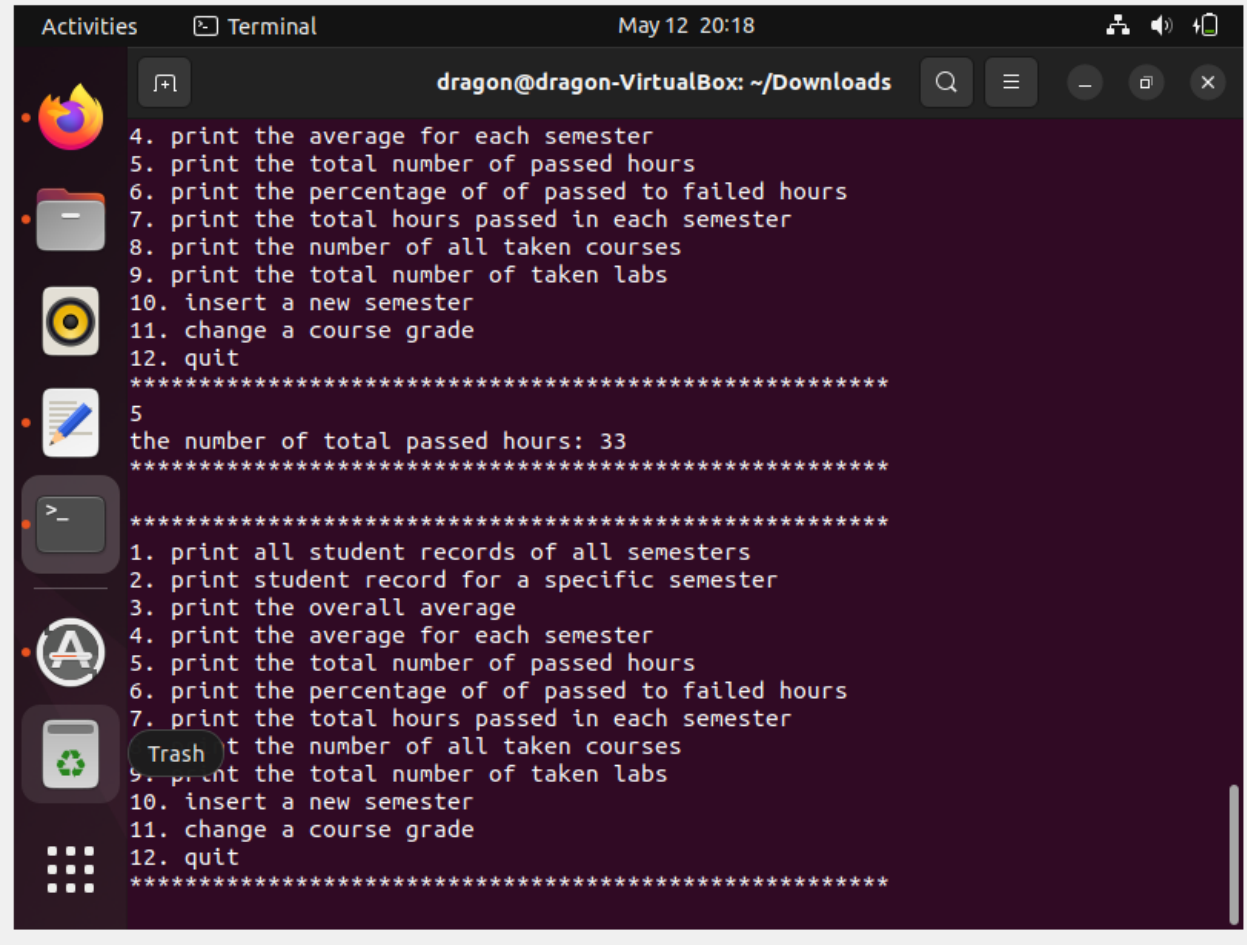
Option 4: print the average for a specific semester

The user specify the year and the semester by entering its number then the average for all marks for that semester will be calculated and printed



Option 5: print the total number of passed hours

This will calculate the sum of all courses which grade is more than 60, which means it is a passed course.



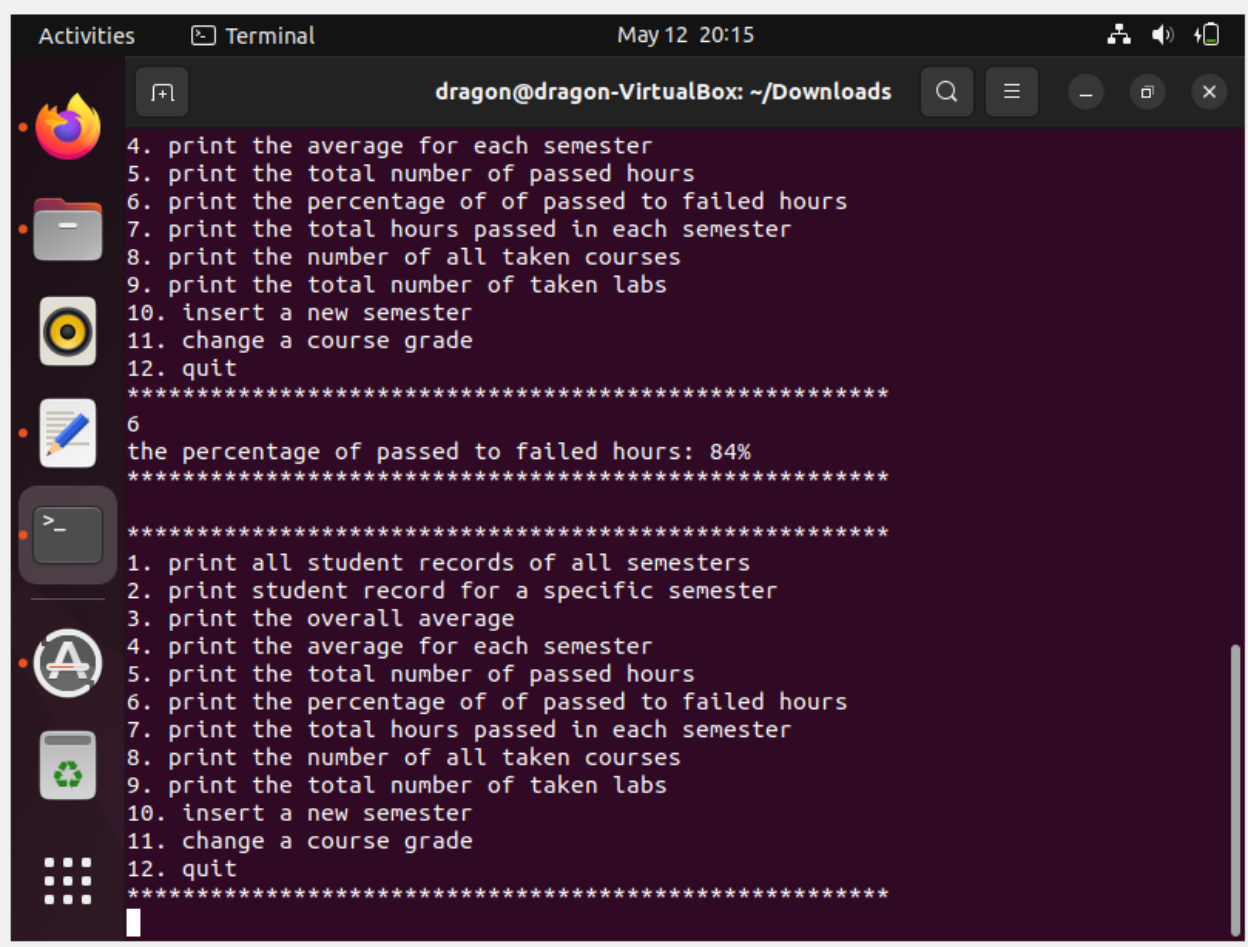
If the year and semester are valid

It will show the list of courses in that semester .

Option 6: print the percentage of the passed hour

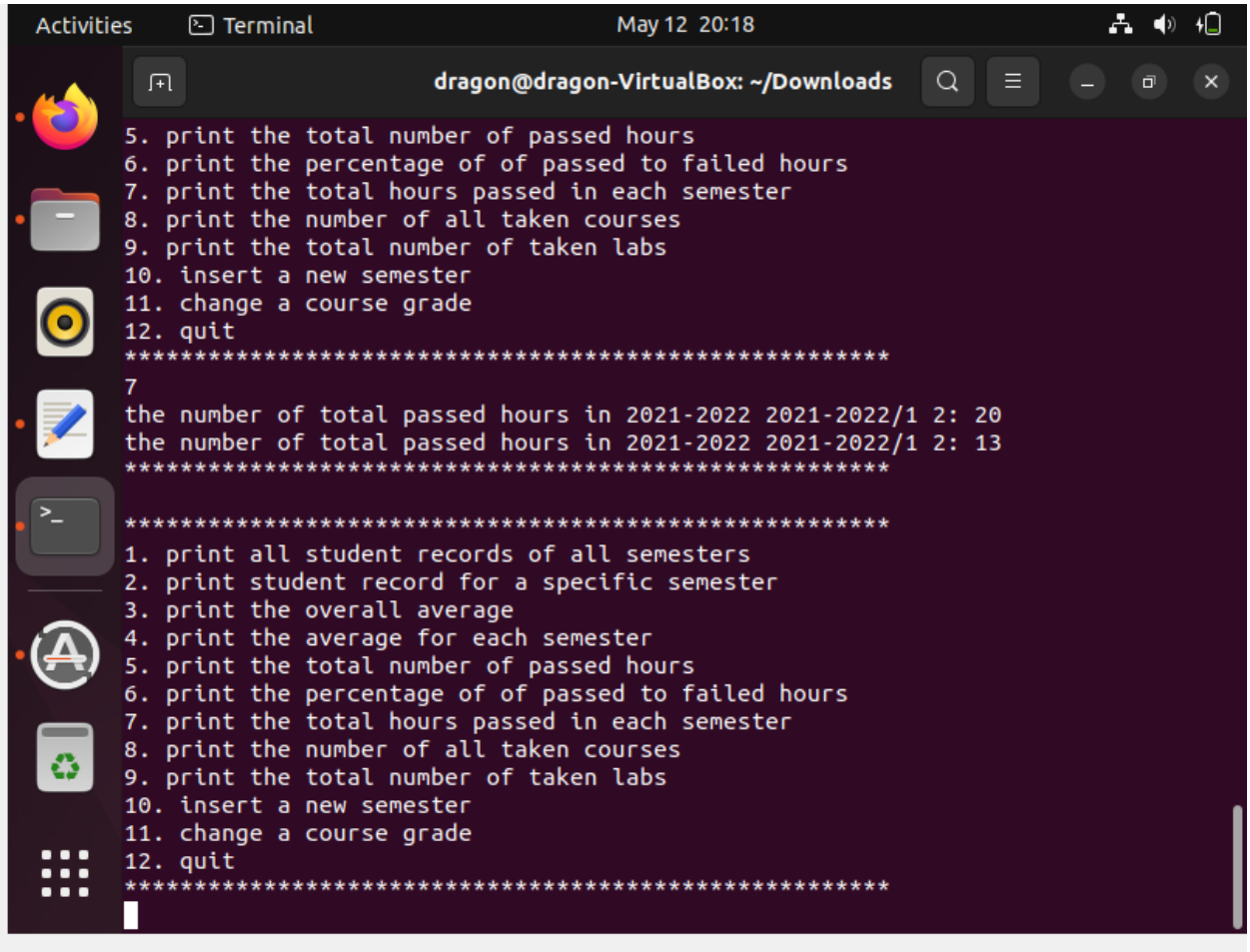
This will calculate the sum of failed hours and passed hours, then the percentage between them

For passed hours

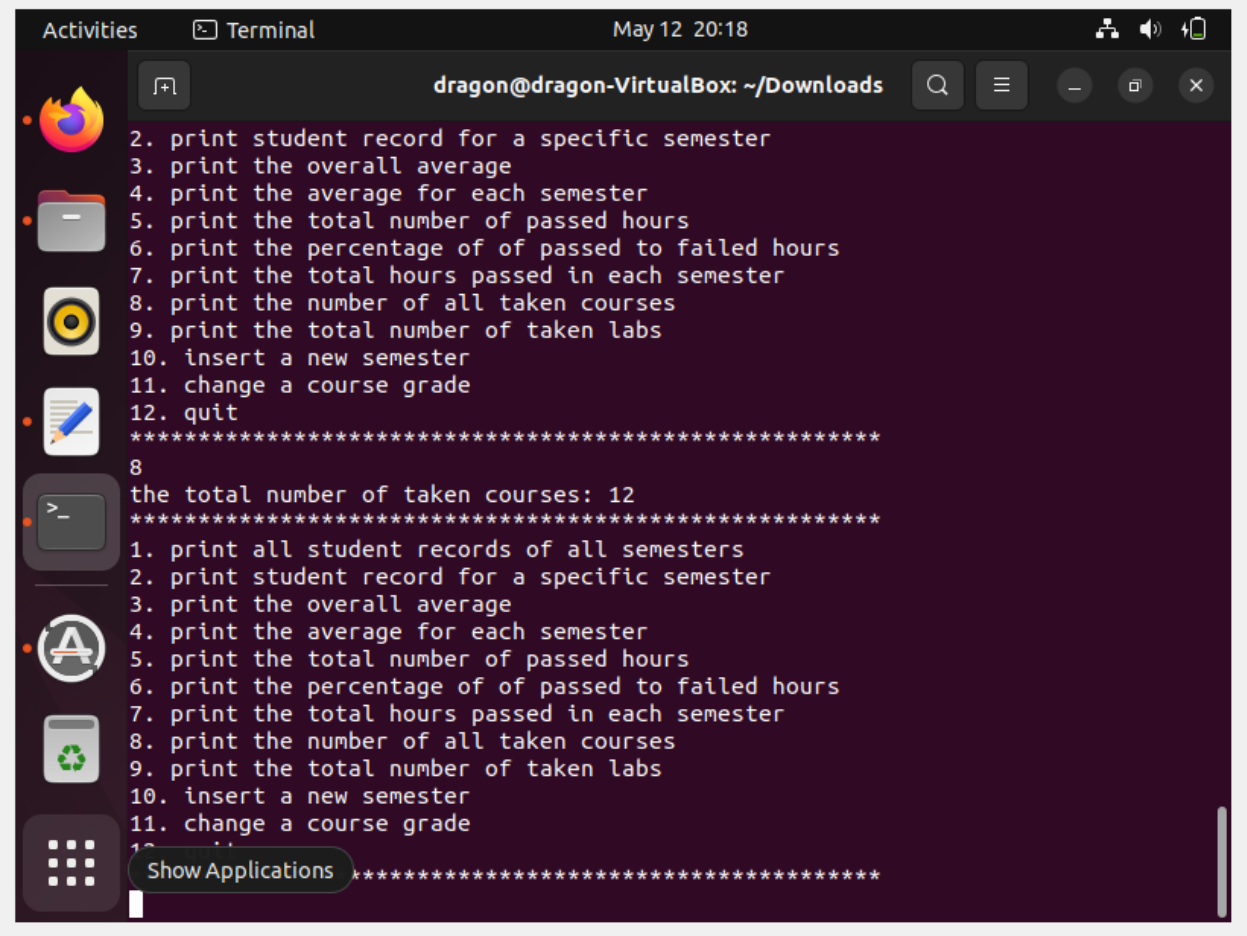


Option 7: print the passed hours in each semester

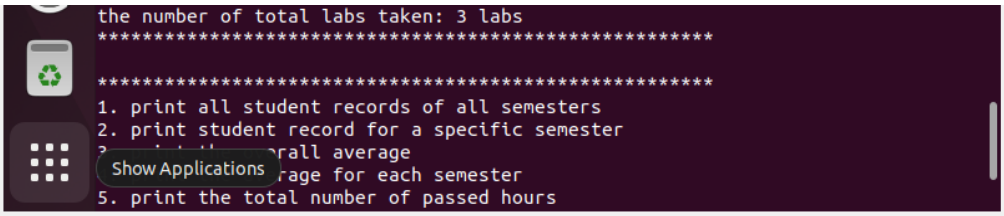
This will print out all semesters and the number of passed hours in each.



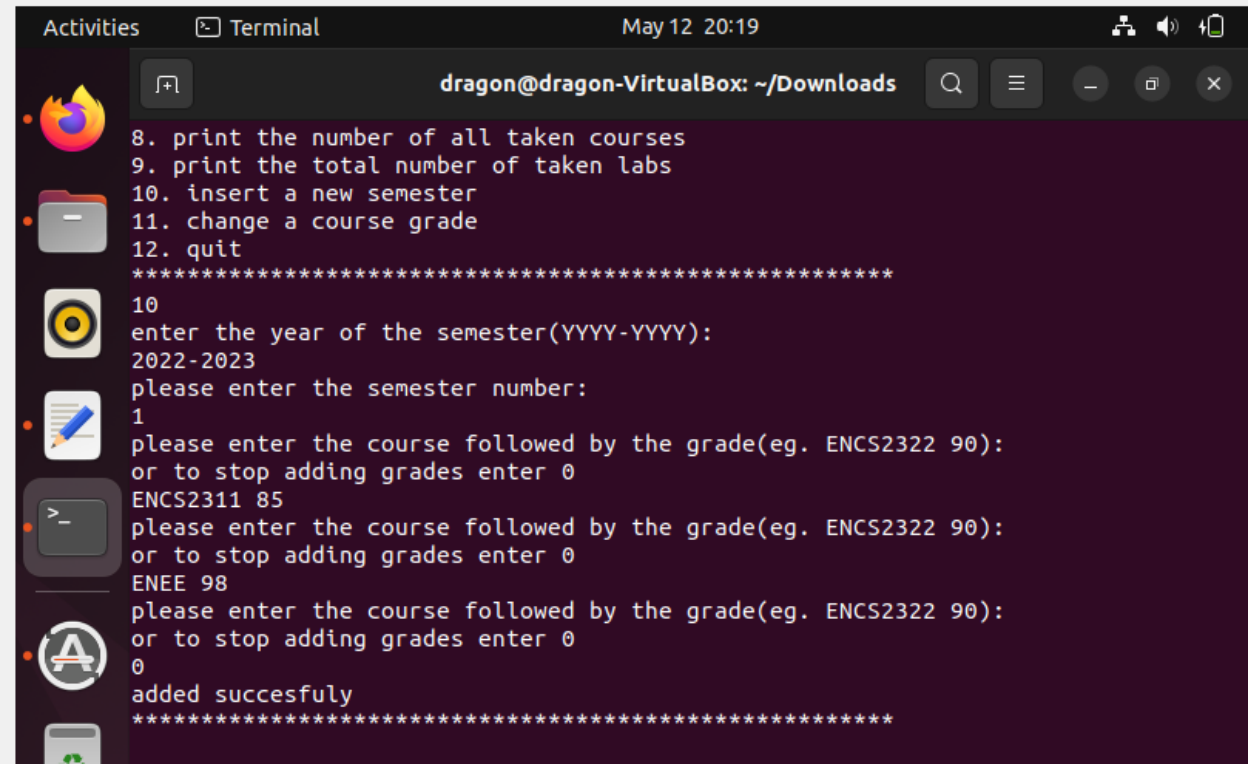
Option 8: number of all taken courses



Option 9: number of taken labs

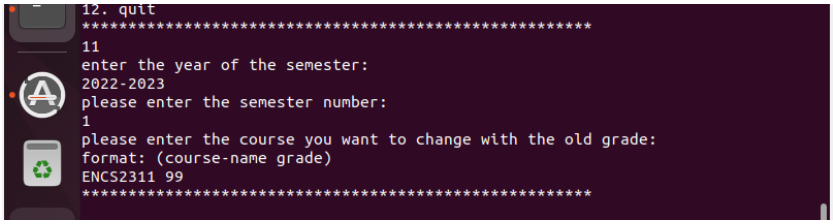


Option 10: add new semester



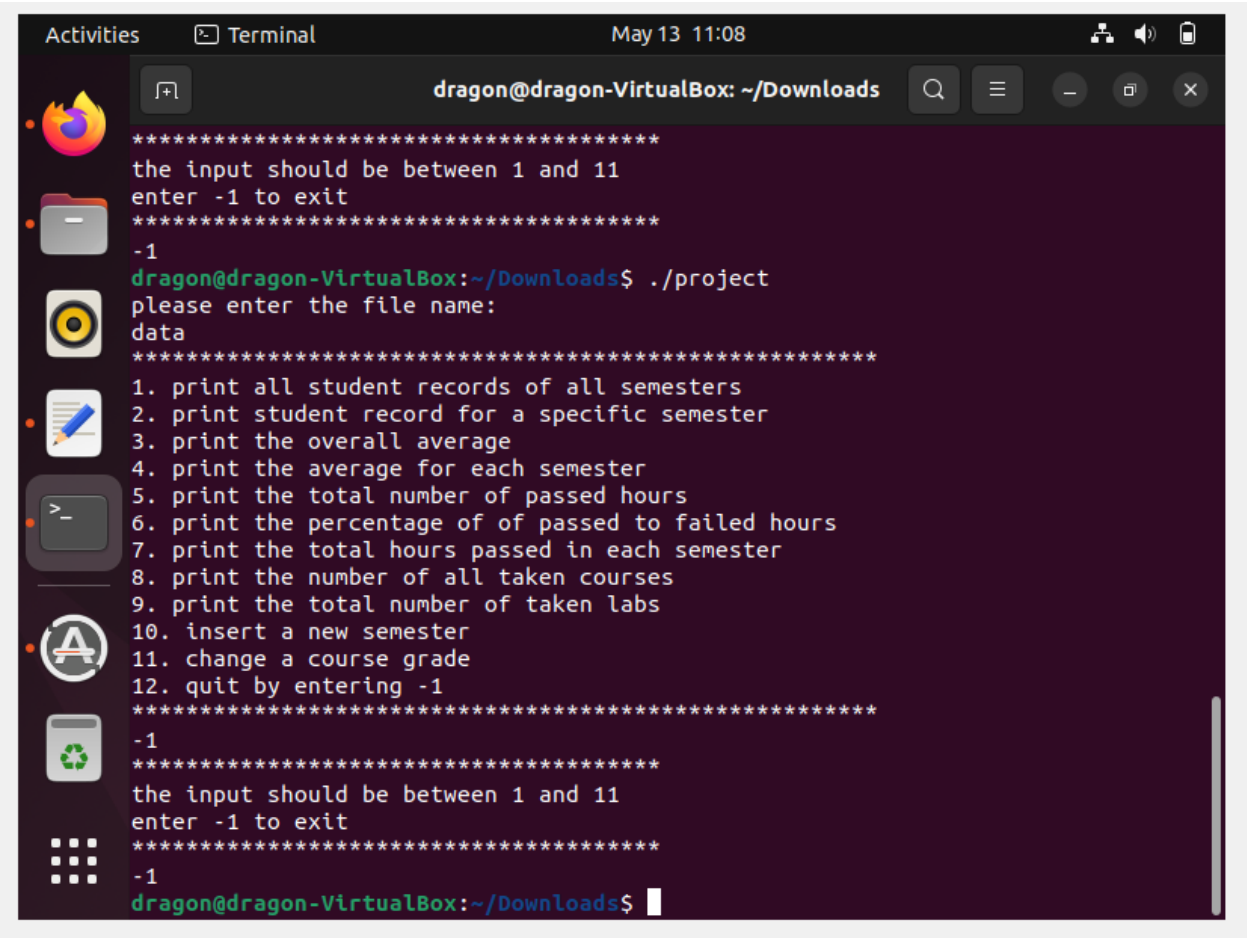
User must specify the year and semester, then he adds as many courses as he wants with its grades, user enters 0 to stop adding more courses.

Option 11: change a course grade



User will be asked to specify year and semester and then course name with the new grade to be edited.

Option 12: quit



The code :

```
#Ahmad Barhoum 1183231
# Abd Alfatah Alkhawaja 1183377
#reading the file name
echo "please enter the file name:"
read file
#check if the file exists
if [ ! -e "$file" ]; then
        echo "File does not exist"
        exit 1
fi
i=0
k=0

declare -A cg

#reading line by line
while read line; do

        #assign the values to the correct variables
        year[$i]=$(cut -d';' -f1 <<< "$line")
        sem[$i]=$(cut -d'/' -f2 <<< "${year[$i]}")
        if [ $((sem[$i])) -lt 1 -o $((sem[$i])) -gt 3 ]; then
                echo "semester can't be other than 1, 2, or 3!"
                echo "update your file!"
                exit 1
        fi
        year[$i]=$(cut -d'/' -f1 <<< "${year[$i]}")
```

```
        cg[$i]=$(cut -d';' -f2 <<< "$line")

        i=$((i+1))

done < $file

while true
do
        #ask user to choose from menu

        echo "*****************************************************"
        echo "1. print all student records of all semesters"
        echo "2. print student record for a specific semester"
        echo "3. print the overall average"
        echo "4. print the average for each semester"
        echo "5. print the total number of passed hours"
        echo "6. print the percentage of of passed to failed hours"
        echo "7. print the total hours passed in each semester"
        echo "8. print the number of all taken courses"
        echo "9. print the total number of taken labs"
        echo "10. insert a new semester"
        echo "11. change a course grade"
        echo "12. quit by entering -1"
        echo "*****************************************************"

        choice=0
        read choice
```

```
while [ $((choice)) -lt 1 -o $((choice)) -gt 11 ]
do
        echo "****************************************"
        echo "the input should be between 1 and 11"
        echo "enter -1 to exit"
        echo "****************************************"
        read choice
        if [  $((choice)) -eq -1 ]; then
                exit 1
        fi

done

if [  $((choice)) -eq 1 ]
then
        for ((j=0;j<i;j++))
        do
                echo "year: ${year[$j]} semester: ${sem[$j]}"
                echo "courses:"
                IFS="," read -r -a arr <<< "${cg[$j]}"
                        for str in "${arr[@]}"
                        do
                                echo $str
                        done
        done
        echo "********************************************************"
        echo ""
```

```
elif [  $((choice)) -eq 2 ]
then

        flag=0
        echo "enter the year of the semester: "
        read y
        echo "please enter the semester number:"
        read s
        while [ $((s)) -lt 1 -o $((s)) -gt 3 ]
        do
                echo "semester can't be other than 1, 2, or 3!"
                echo "try again"
                read s
        done

        for ((j=0;j<i;j++))
        do
                if [[ "${year[$j]}" == "$y" && "$s" == "${sem[$j]}" ]]
                then

                echo "year: ${year[$j]} semester: ${sem[$j]}"
                echo "courses:"
                IFS="," read -r -a arr <<< "${cg[$j]}"
                        for str in "${arr[@]}"
                        do
                                echo $str
                        done
                        flag=1
```

```
                fi
        done
        if [[ $((flag)) -eq 0 ]]
        then
                echo "the year or semester doesn't exist"
        fi

        echo "*********************************************************"
        echo ""

elif [  $((choice)) -eq 3 ]
then
        num=0
        sum=0
        for ((j=0;j<i;j++))
        do
                IFS="," read -r -a arr <<< "${cg[$j]}"
                        for str in "${arr[@]}"
                        do

                                tmp2=$(cut -d' ' -f3 <<< "$str")
                                tmp2=$(tr -d ' ' <<< "$tmp2")
                                if [[ "$tmp2" == "FA" ]]
                                then
                                        tmp2="50"
                                elif [[ "$tmp2" == "I" ]]
                                then
                                        tmp2="0"
```

```
                num=$((num-1))

            elif [[ "tmp2" == "F" ]]
            then
                tmp2="55"
            fi
            sum=$(($((tmp2))+sum))
            num=$((num+1))
        done
    done
    avg=$((sum/num))
    echo "the average is: $avg"

    echo "************************************************************"
    echo ""

elif [  $((choice)) -eq 4 ]
then

    for ((j=0;j<i;j++))
    do

        echo "year: ${year[$j]} semester: ${sem[$j]}"
        num=0
        sum=0
        IFS="," read -r -a arr <<< "${cg[$j]}"
            for str in "${arr[@]}"
            do
```

```
                tmp2=$(cut -d' ' -f3 <<< "$str")
                tmp2=$(tr -d ' ' <<< "$tmp2")
                if [[ "$tmp2" == "FA" ]]
                then
                        tmp2="50"
                elif [[ "$tmp2" == "I" ]]
                then
                        tmp2="0"
                        num=$((num-1))

                elif [[ "tmp2" == "F" ]]
                then
                        tmp2="55"
                fi
                sum=$(($((tmp2))+sum))
                num=$((num+1))
        done
        flag=1
    echo "the average: $((sum/num))"
done
echo "*****************************************************"
echo ""

elif [  $((choice)) -eq 5 ]
then
        num=0
        for ((j=0;j<i;j++))
        do
```

```
                IFS="," read -r -a arr <<< "${cg[$j]}"
                        for str in "${arr[@]}"
                        do
                                tmp3=$(cut -d' ' -f2 <<< "$str")
                                tmp3=$(tr -d ' ' <<< "$tmp3")
                                tmp3=$(cut -c 6 <<< "$tmp3")
                                tmp2=$(cut -d' ' -f3 <<< "$str")
                                tmp2=$(tr -d ' ' <<< "$tmp2")
                                if [[ "$tmp2" == "FA" ]]
                                then
                                        continue
                                elif [[ "$tmp2" == "I" ]]
                                then
                                        continue

                                elif [[ "tmp2" == "F" ]]
                                then
                                        continue
                                fi
                                num=$(($((tmp3))+num))
                        done
        done
        echo "the number of total passed hours: $num"

        echo "*********************************************************"
        echo ""

elif [  $((choice)) -eq 6 ]
```

```
then
        num=0
        fnum=0
        for ((j=0;j<i;j++))
        do
                IFS="," read -r -a arr <<< "${cg[$j]}"
                        for str in "${arr[@]}"
                        do
                                tmp3=$(cut -d' ' -f2 <<< "$str")
                                tmp3=$(tr -d ' ' <<< "$tmp3")
                                tmp3=$(cut -c 6 <<< "$tmp3")
                                tmp2=$(cut -d' ' -f3 <<< "$str")
                                tmp2=$(tr -d ' ' <<< "$tmp2")
                                if [[ "$tmp2" == "FA" ]]
                                then
                                        fnum=$(($((tmp3))+fnum))
                                elif [[ "$tmp2" == "I" ]]
                                then
                                        fnum=$(($((tmp3))+fnum))

                                elif [[ "tmp2" == "F" ]]
                                then
                                        fnum=$(($((tmp3))+fnum))
                                else
                                        num=$(($((tmp3))+num))
                                fi
                        done
        done
```

```
        total=$((fnum+num))
        num=$((num*100))
        echo "the percentage of passed to failed hours: $((num/total))%"
        echo "**********************************************************"
        echo ""

elif [  $((choice)) -eq 7 ]
then

        for ((j=0;j<i;j++))
        do

                num=0
                IFS="," read -r -a arr <<< "${cg[$j]}"
                        for str in "${arr[@]}"
                        do
                                tmp3=$(cut -d' ' -f2 <<< "$str")
                                tmp3=$(tr -d ' ' <<< "$tmp3")
                                tmp3=$(cut -c 6 <<< "$tmp3")
                                tmp2=$(cut -d' ' -f3 <<< "$str")
                                tmp2=$(tr -d ' ' <<< "$tmp2")
                                if [[ "$tmp2" == "FA" ]]
                                then
                                        continue
                                elif [[ "$tmp2" == "I" ]]
                                then
                                        continue
```

```
                elif [[ "tmp2" == "F" ]]
                then
                        continue
                fi
                num=$(($((tmp3))+num))
            done
        echo "the number of total passed hours in ${year[@]}/${sem[@]}: $num"
    done
    echo "*********************************************************"
    echo ""

elif [  $((choice)) -eq 8 ]
then
    num=0
    for ((j=0;j<i;j++))
    do

        IFS="," read -r -a arr <<< "${cg[$j]}"
            for str in "${arr[@]}"
            do
                num=$((num+1))
            done

    done
    echo "the total number of taken courses: $num"

elif [  $((choice)) -eq 9 ]
then
```

```
        num=0
        for ((j=0;j<i;j++))
        do
                IFS="," read -r -a arr <<< "${cg[$j]}"
                        for str in "${arr[@]}"
                        do
                                tmp3=$(cut -d' ' -f2 <<< "$str")
                                tmp3=$(tr -d ' ' <<< "$tmp3")
                                tmp3=$(cut -c 6 <<< "$tmp3")
                                echo $tmp3
                                if [[ $((tmp3)) -eq 1 ]]
                                then
                                        num=$((num+1))
                                fi
                        done
        done
        echo "the number of total labs taken: $num labs"
        echo "**********************************************************"
        echo ""

elif [  $((choice)) -eq 10 ]
then

        flag=0
        echo "enter the year of the semester(YYYY-YYYY): "
        read y
        num1=$(cut -d'-' -f1 <<< "$y")
        num2=$(cut -d'-' -f2 <<< "$y")
```

```
while [[ $(($((num1))+$((num2)))) -lt 4030 || $(($((num1))+$((num2)))) -gt 4050 ]]
do
	echo "the year is not logical"
	echo "please try again(to exit enter 0)"
	read y
	num1=$(cut -d'-' -f1 <<< "$y")
	num2=$(cut -d'-' -f2 <<< "$y")
	if [[ "$y" == "0" ]]
	then
		flag=1
		break
	fi
done
if [[ $((flag)) -eq 1 ]]
then
	break
fi

echo "please enter the semester number:"
read s
while [ $((s)) -lt 1 -o $((s)) -gt 3 ]
do
	echo "semester can't be other than 1, 2, or 3!"
	echo "try again"
	read s
done

input="AA"
```

```
tmp4=" "
flag=0
while [[ "$input" != "0" ]]
do

        echo "please enter the course followed by the grade(eg. ENCS2322 90): "
        echo "or to stop adding grades enter 0"
        read input
        if [[ $((flag)) -eq 1 && "$input" != "0" ]]
        then
                tmp4+=", "
        fi
        flag=1
        if [[ "$input" != "0" ]]
        then
                tmp4+=$input

        fi
done
year[$i]="$y"
sem[$i]="$s"
cg[$i]="$tmp4"
echo "added succesfuly"
echo "**********************************************************"
echo ""
i=$((i+1))

elif [  $((choice)) -eq 11 ]
```

```
then

        flag=0
        echo "enter the year of the semester: "
        read y
        echo "please enter the semester number:"
        read s
        while [ $((s)) -lt 1 -o $((s)) -gt 3 ]
        do
                echo "semester can't be other than 1, 2, or 3!"
                echo "try again"
                read s
        done

        for ((j=0;j<i;j++))
        do
                if [[ "${year[$j]}" == "$y" && "$s" == "${sem[$j]}" ]]
                then

                        flag=1
                        break
                fi
        done
        if [[ $((flag)) -eq 0 ]]
        then
                echo "the year or semester doesn't exist"
                continue
        fi
```

```
            echo "please enter the course you want to change with the old grade:"
            echo "format: (course-name grade)"
            read old

            if [[ "${cg[$j]}" == *"$old"* ]]
            then

                echo "please enter the course you want to change with the new grade:"
                echo "format: (course-name grade)"
                read sub

                cg[$j]=${cg[$j]/"$old"/"$sub"}
            fi
            echo "**********************************************************"
            echo ""

        else
            exit 1
        fi

done
```

Conclusion

In this project we learned how to deal with data from file, to do a simple university registration system, we have practiced shell scripting.

During writing the script we tried to take in consideration all possible cases