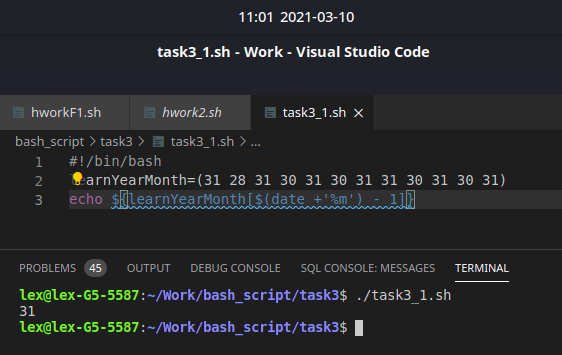
# Exercise 1

#### Develop a script that prints information about the current month. The script should print the number of days of the current month.

learnYearMonth=(31 28 31 30 31 30 31 31 30 31 30 31)

echo ${learnYearMonth[$(date +'%m') - 1]}



# Exercise 2

#### Do the same, using a case statement and an alternative use of the date command.

#!/bin/bash

learnYearMonth=(31 28 31 30 31 30 31 31 30 31 30 31)

case $(date +'%m') in

'1')

echo $learnYearMonth[0]

;;

'2')

echo ${learnYearMonth[1]}

;;

'3')

echo ${learnYearMonth[2]}

;;

'4')

echo ${learnYearMonth[3]}

;;

'5')

echo ${learnYearMonth[4]}

;;

'6')

echo ${learnYearMonth[5]}

;;

'7')

echo ${learnYearMonth[6]}

;;

'8')

echo ${learnYearMonth[7]}

;;

'9')

echo ${learnYearMonth[8]}

;;

'10')

echo ${learnYearMonth[9]}

;;

'11')

echo ${learnYearMonth[10]}

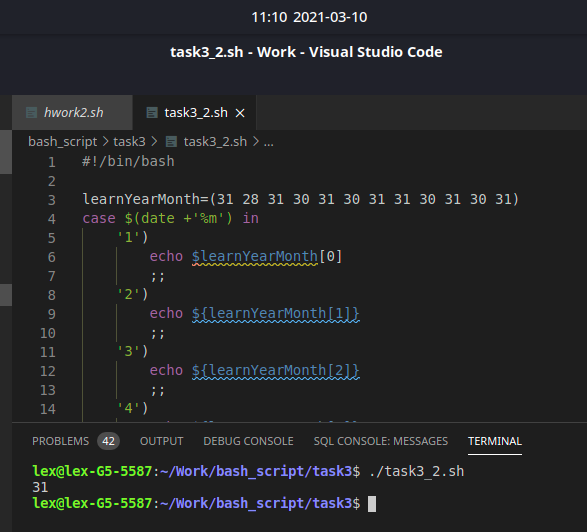
;;

\*)

echo ${learnYearMonth[11]}

;;

esac



# Exercise 3

#### Develop a script called 03-which-daemon.sh that checks if the httpd and init daemons are running on your system. If an httpd is running, the script should print a message like, "This machine is running a web server." Use ps to check on processes

for t in $(ps -ef | grep cups | awk $'{print $8}')

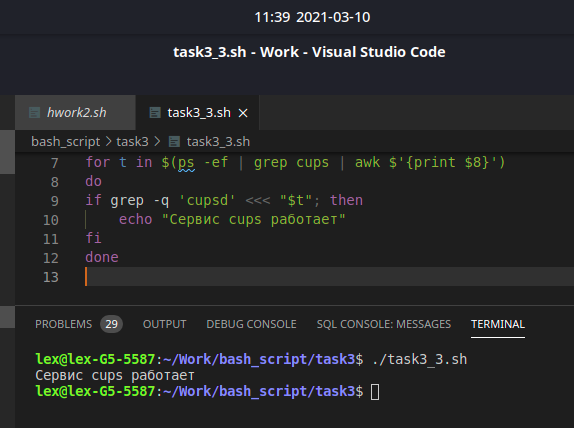
do

if grep -q 'cupsd' <<< "$t"; then

echo "Сервис cups работает"

fi

done



# Exercise 4

#### Write a script that executes the command cat /etc/shadow. If the command return a 0 exit status, report Command succeeded and exit with a 0 exit status. If the command returns a non-zero exit status, report Command failed and exit with a 1 exit status.

cat /etc/shadow &>/dev/null

if [ $? -eq 0 ]

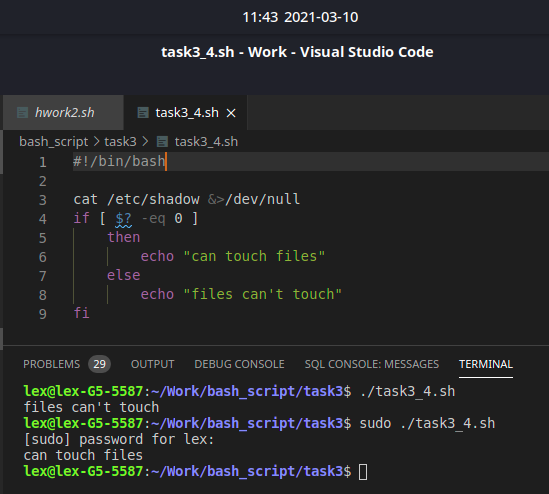
then

echo "can touch files"

else

echo "files can't touch"

fi



# Exercise 5

#### Modify following script:

#### Requirements:

Script should print maximum number from generated list

Script should print minimal number from generated list

Script should print sum of all generated numbers

function rand() {

echo $(($RANDOM % ${1}))

}

for item in `seq 1 10`

do

value="$(rand 100)"

echo "[$item] => ${value}"

done 1>./test

#в данном упражнении я скопировал вывод работы функции из потока вывода

#во временный файл test.txt

cat ./test

#вывод информации из файла (вывод как в примере)

awk '{print $3}' ./test > ./test1

#преобразование файла с последущим сохранением в файл test1.txt (т.к. файл сожержит лишние символы)

readarray test\_array < ./test1

#занесение данных из файла test1.txt в массив test\_array

rm test test1

#удаление временных файлов

max=test\_array[0] #поиск max, min, sum

min=test\_array[0]

sum=0

for i in ${test\_array[@]}

do

if [[ $i -gt $max ]] ; then

max=$i

fi

if [[ $i -lt $min ]] ; then

min=$i

fi

sum=$(($sum+$i))

done

echo "max=$max"

echo "min=$min"

echo "sum=$sum"

