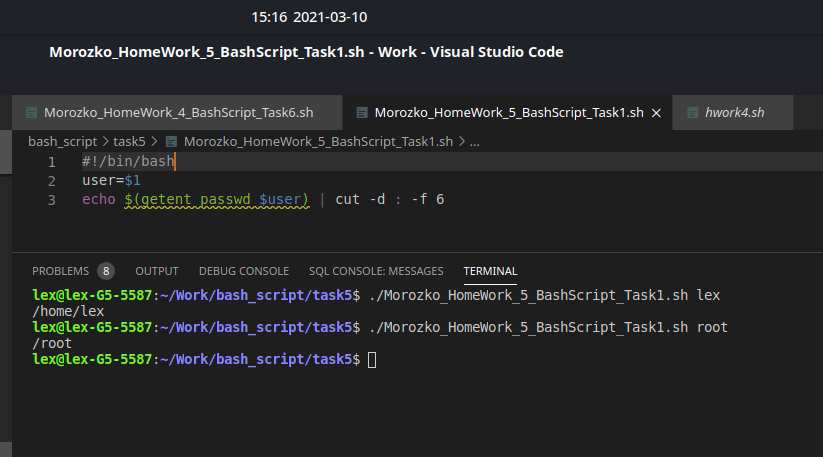
#Exercise 1

#### Write a Bash shell script program that takes a login name as an argument and outputs the home directory of the user.

user=$1

echo $(getent passwd $user) | cut -d : -f 6



# Exercise 2

#### If you are 18 or over, you may go to the party. If you aren't but you have a letter from your parents you may go but must be back before midnight. Otherwise you cannot go.

Requirements:

Script should expect 2 input parameters: the first - age, the second - yes/no (depends on whether you have letter from parents or don't)

Script should generate appropriate output.

ages=$1

letter=$2

echo ${ages}

echo ${letter}

if (( ${ages} > 17 ))

then

echo "You may go."

else

if [ ${letter} = "yes" ]

then

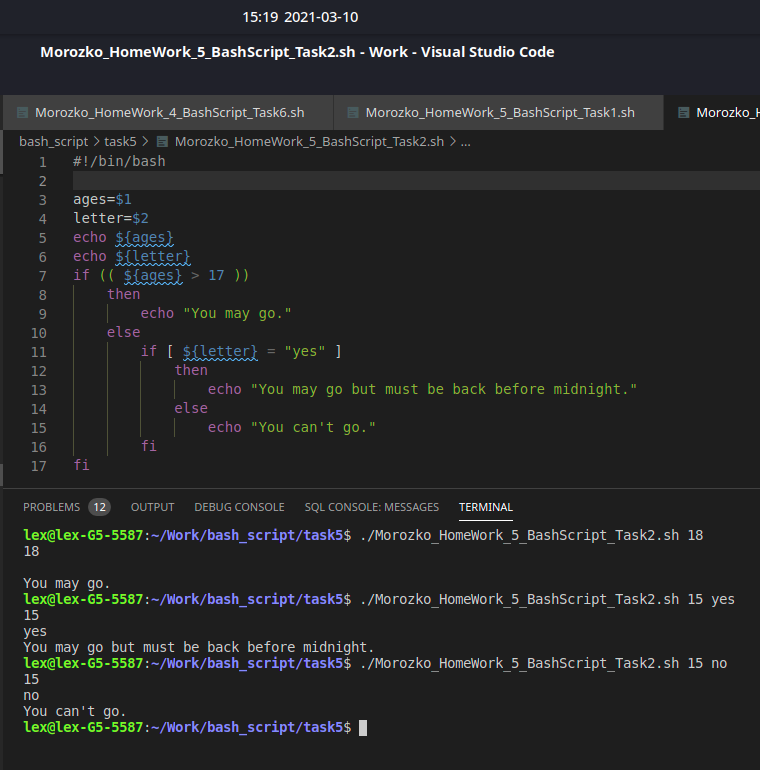
echo "You may go but must be back before midnight."

else

echo "You can't go."

fi

fi



# Exercise 3

#### Write a Bash shell script which prints out information about file or directory, provided in argument.

Requirements:

Script should take the name of a file or directory as an argument and reports if the file is a directory, regular file or other

Script should print out read, write and execute permission applied on the file or directory

Use a sequence of if statements on the file name to determine the information

Hint: try to use stat filename command

direct=$1

for t in $(ls -l ${direct} | awk 'NR > 1 {print $9}')

do

if [ -d "$1$t" ]

then

echo "File: '$1/$t'"

echo "Type: directory"

echo "Permission: ("$(stat -c '%a' $1/$t)'/'$(ls -ld $1/$t | awk '{print $1}')")"

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

else echo "File: '$1/$t'"

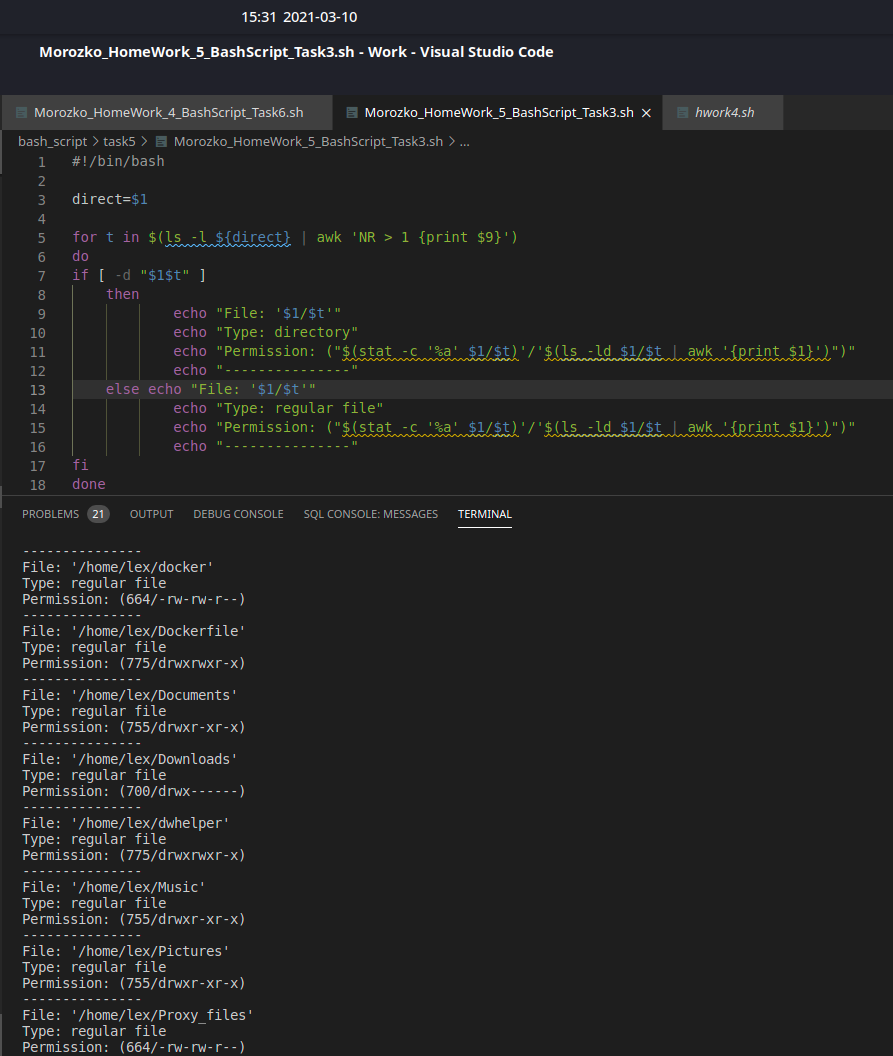
echo "Type: regular file"

echo "Permission: ("$(stat -c '%a' $1/$t)'/'$(ls -ld $1/$t | awk '{print $1}')")"

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

fi

done



# Exercise 4

#### Write a Bash shell script program that will read a list of numbers from arguments and the output is the sum and the product of the numbers.

file=$@

read -a test\_arr <<< $@

sum=0

prod=1

for i in ${test\_arr[@]}; do

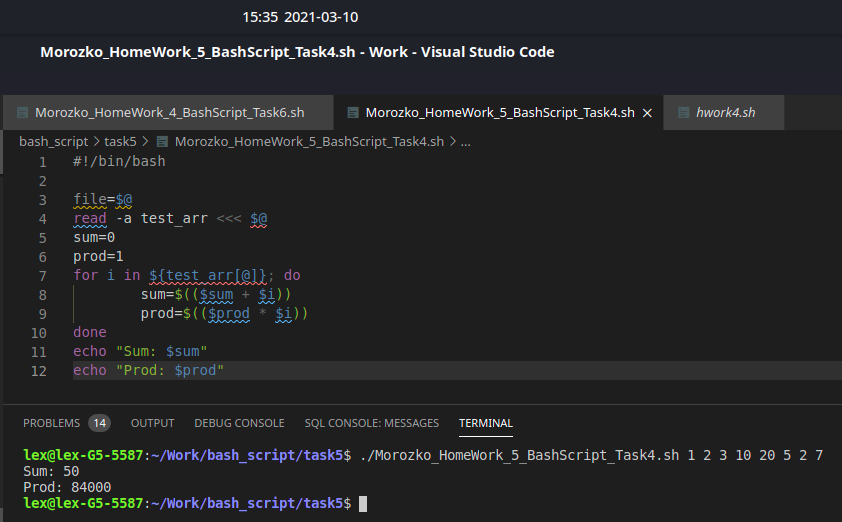
sum=$(($sum + $i))

prod=$(($prod \* $i))

done

echo "Sum: $sum"

echo "Prod: $prod"



# Exercise 5

#### Write a Bash shell script program that will read a list of numbers from arguments and the output is the min and the max element of the numbers.

file=$@

read -a test\_arr <<< $@

min=${test\_arr[0]}

max=${test\_arr[0]}

for i in ${test\_arr[@]};do

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

max=$i

fi

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

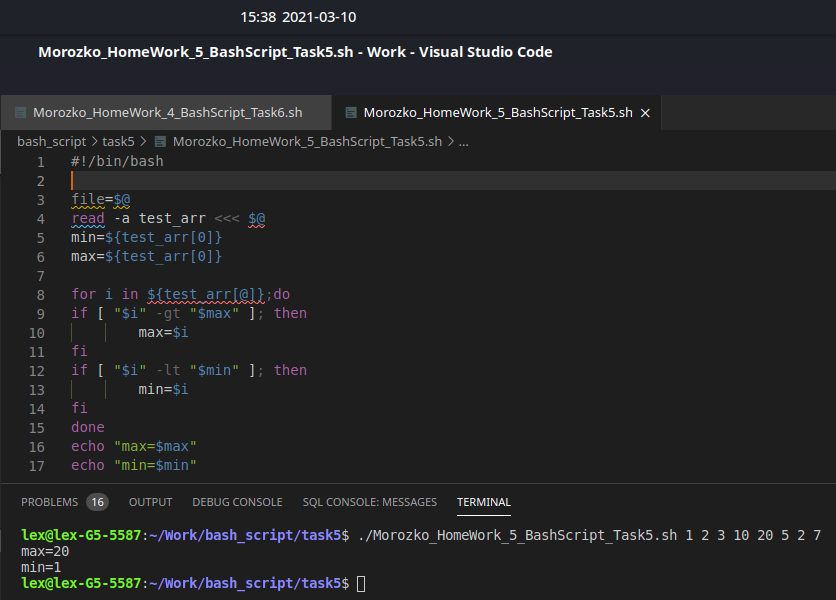
min=$i

fi

done

echo "max=$max"

echo "min=$min"



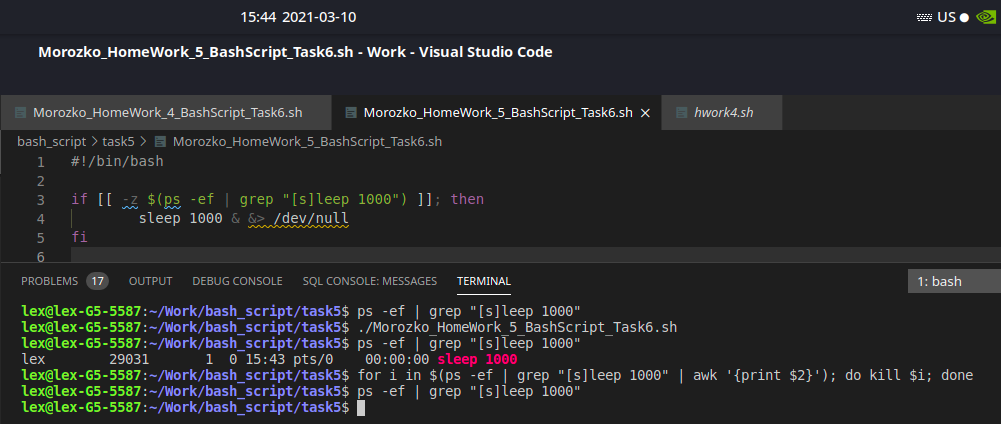
# Exercise 6

#### Develop a script which will be check whether sleep 1000 ran in background. If sleep 1000 haven’t run in background – do it:

if [[ -z $(ps -ef | grep "[s]leep 1000") ]]; then

sleep 1000 & &> /dev/null

fi



# Exercise 7

**Script should print its’ full path and its’ name**

pwd

echo $(basename $0)

