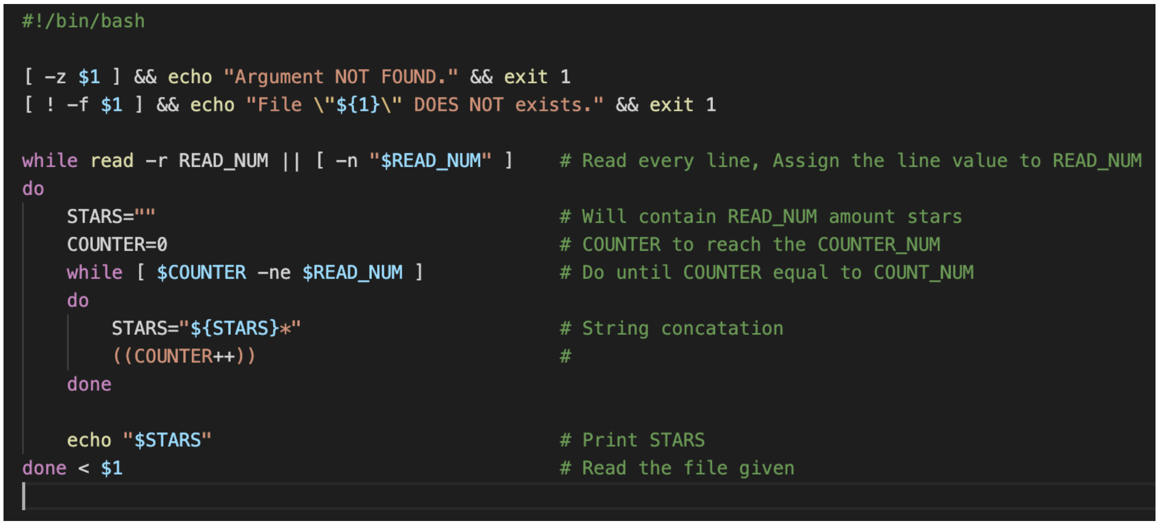
Ceng322
PROGRAMMING ASSIGNMENT 1
Umut YILDIZ - 260201028

- Script 1)

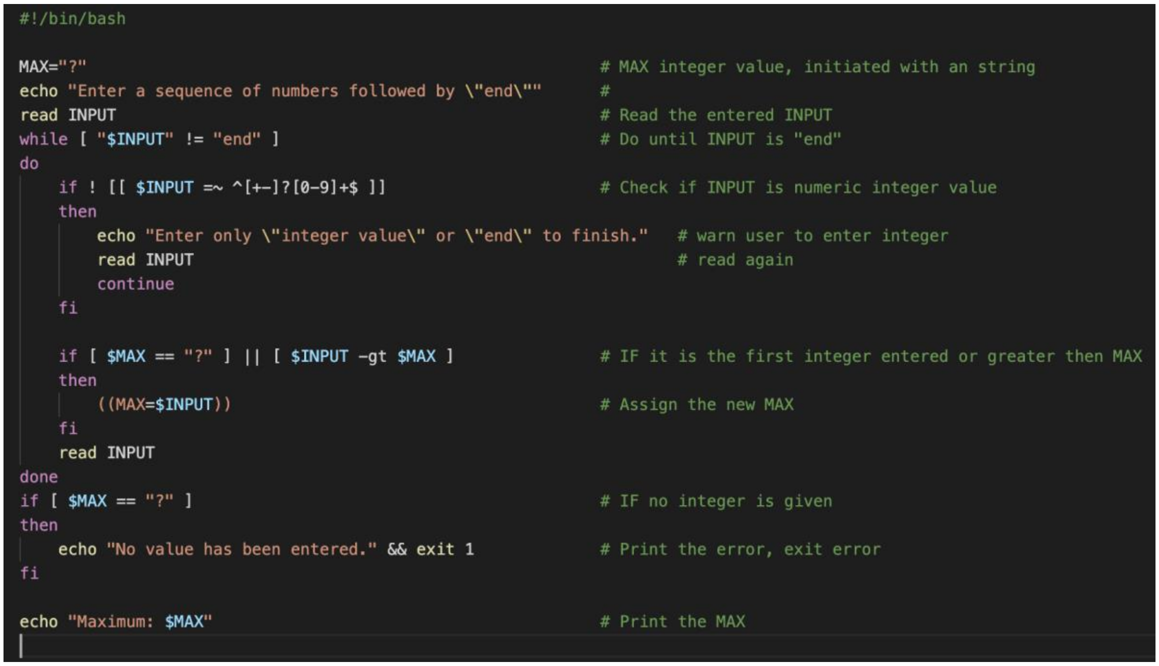
- Before starting to run, I check if the given file name as argument exists.
- I assumed that in the given file’s every line consists of only one positive integer value.
- Inner while concatenate read number of stars for later use.
- Finally prints the stars for each read line.



```
#!/bin/bash

[ -z $1 ] && echo "Argument NOT FOUND." && exit 1
[ ! -f $1 ] && echo "File \"${1}\" DOES NOT exists." && exit 1

while read -r READ_NUM || [ -n "$READ_NUM" ]    # Read every line, Assign the line value to READ_NUM
do
    STARS=""                                    # Will contain READ_NUM amount stars
    COUNTER=0                                   # COUNTER to reach the COUNTER_NUM
    while [ $COUNTER -ne $READ_NUM ]            # Do until COUNTER equal to COUNT_NUM
    do
        STARS="${STARS}*"                       # String concatation
        ((COUNTER++))                           #
    done

    echo "$STARS"                               # Print STARS
done < $1                                       # Read the file given
```

- Script 2)

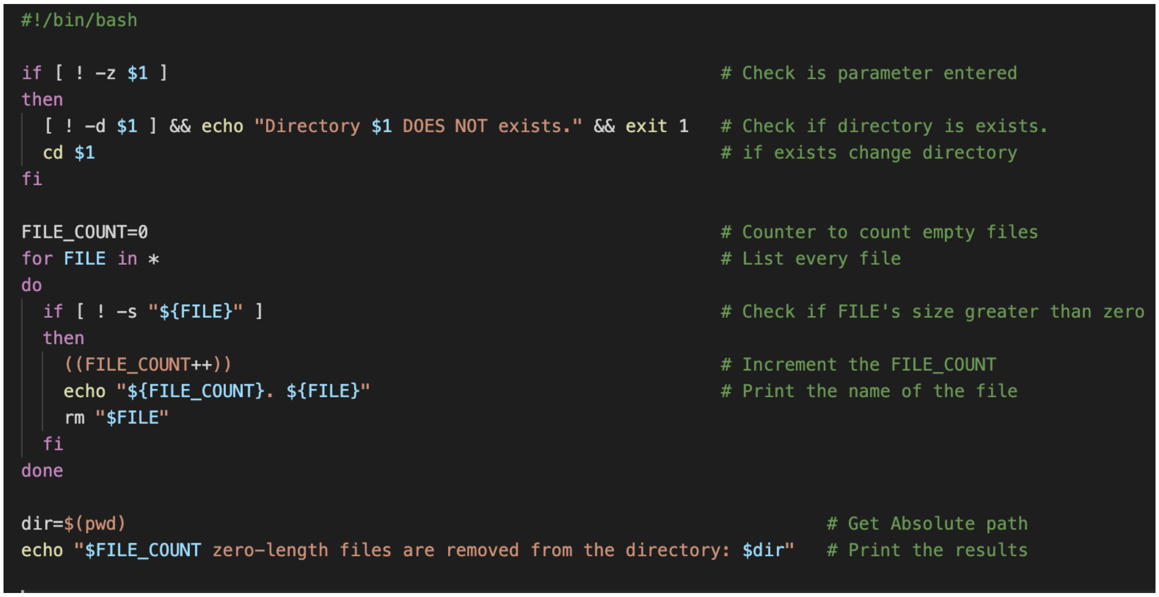
- Initialize the maximum value as non-numeric value to recognize the first input.
- If no numbers are entered, then max will still be “?”. Then we can check this condition easily.
- First if checks whether the entered value is integer. If not ask user to enter an integer or “end”.
- Second if checks whether the entered value is first or bigger than previous max value.



```
#!/bin/bash

MAX="?"                                                     # MAX integer value, initiated with an string
echo "Enter a sequence of numbers followed by \"end\""      #
read INPUT                                                  # Read the entered INPUT
while [ "$INPUT" != "end" ]                                 # Do until INPUT is "end"
do
    if ! [[ $INPUT =~ ^[+-]?[0-9]+$ ]]                      # Check if INPUT is numeric integer value
    then
        echo "Enter only \"integer value\" or \"end\" to finish."   # warn user to enter integer
        read INPUT                                                  # read again
        continue
    fi

    if [ $MAX == "?" ] || [ $INPUT -gt $MAX ]               # IF it is the first integer entered or greater then MAX
    then
        ((MAX=$INPUT))                                      # Assign the new MAX
    fi
    read INPUT
done
if [ $MAX == "?" ]                                          # IF no integer is given
then
    echo "No value has been entered." && exit 1             # Print the error, exit error
fi

echo "Maximum: $MAX"                                        # Print the MAX
```

- Script 3)

- Start by checking if argument is given. If given, then check if valid. If valid change directory.
- If the argument is not given, continue working on the same workspace.
- Loop through every file and check if there exists a file whose size is not greater than zero.



```
#!/bin/bash

if [ ! -z $1 ]                                                    # Check is parameter entered
then
  [ ! -d $1 ] && echo "Directory $1 DOES NOT exists." && exit 1   # Check if directory is exists.
  cd $1                                                           # if exists change directory
fi

FILE_COUNT=0                                                      # Counter to count empty files
for FILE in *                                                     # List every file
do
  if [ ! -s "${FILE}" ]                                           # Check if FILE's size greater than zero
  then
    ((FILE_COUNT++))                                              # Increment the FILE_COUNT
    echo "${FILE_COUNT}. ${FILE}"                                 # Print the name of the file
    rm "$FILE"
  fi
done

dir=$(pwd)                                                                  # Get Absolute path
echo "$FILE_COUNT zero-length files are removed from the directory: $dir"   # Print the results
```