Instructions:

PART A: WRITE BASH SHELL SCRIPT CODE

Write the answer to each question below the question in the space provided.

1. Write a Bash shell script that clears the screen and displays the text **Hello World** on the screen.

```
->
#!/bin/bash
clear
echo "Hello World"
```

What **permissions** are required to run this Bash shell script?
-> **chmod +x script_name.sh**

What are the different **methods** to run this Bash shell script from the command line?

```
->
   ./script_name.sh (if in the same directory)
   /path/to/script/script_name.sh
```

2. Write a Bash shell script that clears the screen, prompts the user for their **full name** and then prompts the user for their **age**, then clears the screen again and welcomes the user by their name and tells them their age.

```
->
#!/bin/bash
clear
echo -n "Enter your full name: "
read fullName
echo -n "Enter your age: "
read age
clear
echo "Welcome, $fullName! Your age is $age."
```

What comments would you add to the above script's contents to properly document this Bash shell script to be understood for those users that would read / edit this Bash shell script's contents?

```
->
#!/bin/bash
# This script takes user information and displays a welcome
message.
# It prompts for the full name and age of the user.
```

3. Write a Bash shell script that will first set the value of a read-only variable called **dogFactor** to **7**. The script will then clear the screen and prompt the user to enter the age of a dog in human years (which will be stored into a variable called **humanYears**).

The script will store in a variable called **dogYears** the value of **humanYears x dogFactor**
The script will then clear the screen a second time and then display the age of the dog in *"dog years"*.

->

```
#!/bin/bash
readonly dogFactor=7
clear
echo -n "Enter the age of a dog in human years: "
read humanYears
dogYears=$((humanYears * dogFactor))
clear
echo "The age of the dog in dog years is: $dogYears"
```

4. Write a Bash shell script that will clear the screen and then display all arguments that were entered after your Bash shell script when it was run.

Also, have the Bash shell script display the
number of arguments that were entered after your Bash shell script.

->

```
#!/bin/bash
clear
echo "Arguments entered after the script: \$1=$1, \$2=$2, \$3=$3"
echo "Number of arguments: $#"
```

PART B: WALK-THRUS

Write the expected output from running each of the following Bash shell scripts
You can assume that these Bash shell script files have execute permissions.
Show your work.

1. **cat walkthru1.bash**

```
#!/usr/bin/bash
word1=”counter”
word2=”clockwise”
echo “The combined word is: $word2$word1”
```

WRITE ROUGH WORK AND OUTPUT FROM ISSUING:
./walkthru1.bash

ROUGH WORK:
->
word1="counter"
word2="clockwise"
echo "The combined word is: clockwisecounter"

OUTPUT:
->
The combined word is: clockwisecounter

2. **cat walkthru2.bash**

```
#!/usr/bin/bash
echo “result1: $1”
echo “result2: $2”
echo “result3: $3”
echo “result 4:”
echo “$*”
```

WRITE ROUGH WORK AND OUTPUT FROM ISSUING:
./walkthru2.bash apple orange banana

ROUGH WORK:
->
./walkthru2.bash apple orange banana

OUTPUT:
->
result1: apple
result2: orange
result3: banana
result 4:

```
apple orange banana
```

3. cat walkthru3.bash

```
#!/usr/bin/bash

for x in 1 2 3 4 5
do

  if [ $((x % 2)) -eq 0 ]
  then
    echo "this"
  else
    echo "that"
  fi

done
```

WRITE ROUGH WORK AND OUTPUT FROM ISSUING:
./walkthru3.bash

ROUGH WORK:
->

```
for x in 1 2 3 4 5
do
 if [ $((x % 2)) -eq 0 ]
 then
  echo "this"
 else
  echo "that"
 fi
done
```

OUTPUT:

➔ that
➔ this
➔ that
➔ this
➔ that