

1. Write a shell script to define any four variables (of different data types) and display them using echo. For example:

myval=45

echo "Value of myval is \$myval"

```
#!/bin/bash
my_integer=41
my_string="Om"
my_float=3.14
my_boolean=true

echo "Value of integer is $my_integer"
echo "Value of string is $my_string"
echo "Value of float is $my_float"
echo "Value of my boolean is $my_boolean"
```

```
~/A41 🕒 1:48:54
$ bash variable_1.sh
Value of integer is 41
Value of string is Om
Value of float is 3.14
Value of my boolean is true
```

2. Accept the age of the user from input. Write a shell script to determine if the user is eligible for voting.

```
#!/bin/bash

echo "Enter your age: "
read age

if [[ "$age" -ge 18 ]]; then
    echo "You are eligible for voting"
else
    echo "You are not eligible for voting"
fi
```

```
~/A41 🕒 2:05:10
$ bash age_2.sh
Enter your age:
18
You are eligible for voting
```

3. Write a shell script that accepts two numbers from users. Display the addition, subtraction, multiplication, division (quotient), remainder results using echo; without and with expr.

```
#!/bin/bash

echo "Enter the two numbers"
read num1 num2

# Using echo

echo "Results using echo:"
echo "Addition      : $((num1 + num2))"
echo "Subtraction    : $((num1 - num2))"
echo "Multiplication : $((num1 * num2))"
echo "Division        : $((num1 / num2))"
echo "Remainder       : $((num1 % num2))"
echo "-----"

# Using expr

echo "Results using expr:"
echo "Addition      : $(expr $num1 + $num2)"
echo "Subtraction    : $(expr $num1 - $num2)"
echo "Multiplication : $(expr $num1 \* $num2)"
echo "Division        : $(expr $num1 / $num2)"
echo "Remainder       : $(expr $num1 % $num2)"
```

```
~/A41 🕒 2:18:30
$ bash arithmetic_3.sh
Enter the two numbers
20 34
Results using echo:
Addition      : 54
Subtraction   : -14
Multiplication : 680
Division      : 0
Remainder     : 20
-----
Results using expr:
Addition      : 54
Subtraction   : -14
Multiplication : 680
Division      : 0
Remainder     : 20
```

4. Write a shell script to determine the maximum of three numbers.

```
#!/bin/bash

echo "Enter the numbers"
read num1 num2

if [ $num1 -gt $num2 ]
then
    echo "1st number max"
elif [ $num1 -lt $num2 ]
then
    echo "2nd number max"
else
    echo "Equal"
fi
```

```
~/A41 🕒 2:23:03
$ bash max_4.sh
Enter the numbers
20 43
2nd number max
```

5. Accept two strings from the user. Write a shell script to check if the two strings are the same or not.

```
#!/bin/bash

echo "Enter the strings"
read str1 str2

if [ $str1 = $str2 ]
then
    echo "Two strings are same"
else
    echo "Two strings are different"
fi
```

```
~/A41 🕒 2:35:55
$ bash compare_string_5.sh
Enter the strings
Om Kadam
Two strings are different
```

6. Write a shell script to validate the course marks using nested if and specify the class. Accept marks for three courses – Physics, Chemistry, Mathematics. Compute the total marks and average marks. If any course has less than 35 marks, display “Failed”. If the average marks is ≥ 75 , display “Distinction”. If the average marks is ≥ 60 but ≤ 75 ,

display "First Class. If the average marks is ≥ 50 but ≤ 60 , display "Second Class. If the average marks is ≥ 35 but ≤ 50 , display "Third Class.

```
#!/bin/bash

calculate_marks() {
    total=$(( $1 + $2 + $3 ))
    average=$(( $total / 3 ))
}

determine_class() {
    if [ $1 -ge 75 ]; then
        echo "Distinction"
    elif [ $1 -ge 60 ]; then
        echo "First Class"
    elif [ $1 -ge 50 ]; then
        echo "Second Class"
    elif [ $1 -ge 35 ]; then
        echo "Third Class"
    else
        echo "Failed"
    fi
}

read -p "Enter marks for Physics: " physics
read -p "Enter marks for Chemistry: " chemistry
read -p "Enter marks for Mathematics: " mathematics
```

```
calculate_marks $physics $chemistry $mathematics

class = $(determine_class $average)

echo "Total marks:    $total"
echo "Average marks:  $average"
echo "Class:          $class"
```

```
$ bash marks_6.sh
Enter marks for Physics: 76
Enter marks for Chemistry: 56
Enter marks for Mathematics: 88
Total marks:    220
Average marks:  73
Class:          First Class
```

7. Display "Linux" i times, where i goes from 1 to 8. Demonstrate using while loop and for loop.

```
#!/bin/bash

# Using while loop
echo "Output using while loop"
i=1
while [ $i -le 8 ]
do
    echo "Linux"
    i=$((i+1))
done
echo " "
echo "Output using for loop"
# Using for loop

for (( i=1; i<=8; i++ ))
do
    echo "Linux"
done
```

```
$ bash loop_7.sh  
Output using while loop  
Linux  
Linux  
Linux  
Linux  
Linux  
Linux  
Linux  
Linux
```

```
Output using for loop  
Linux  
Linux  
Linux  
Linux  
Linux  
Linux  
Linux  
Linux
```

8. Write an infinite loop using for loop (use CTRL + C to exit).

```
#!/ /bin/bash  
  
for (( ;; ))  
do  
    echo "This loop with run infinite times"  
done
```

```
This loop with run infinite times
This loop with run infinite times
This loop with run infinite times
This loop with run infinite times
This loop with run infinite times
This loop with run infinite times
This loop with run infinite times
This loop with run infinite times
^C
```

~/A41 🕒 13:55:14

9. Write a shell script to print a number in reverse order using a while loop. It should support the following requirements. The script should accept the input from the command line. If you don't input any data, then display an error message to execute the script correctly.

Hint:

- a. Suppose the input number is n.
- b. Set reverse to 0 and digit to 0 (i.e., rev=0, digit=0).
- c. The expression $(n \% 10)$ will give the single leftmost digit i.e., digit.
- d. To reverse the number, use this expression $rev * 10 + digit$.
- e. Decrease the input number (n) using $n / 10$.
- f. If n is greater than 0, then go to step no. 3. Else, execute the step no. g.
- g. Print the result.


```

#!/bin/bash

read -p "Enter number: " number

reverse=0
digit=0

while [ $number -gt 0 ]; do
    digit=$(( $number % 10 ))
    reverse=$(( $reverse * 10 + $digit ))
    number=$(( $number / 10 ))
done

echo "Reversed number: $reverse"

```

```

$ bash reverse_9.sh
Enter number: 12345
Reversed number: 54321

```

10. Write a shell script to reverse a user-defined string. Hint: use pipe operator and rev.

```

#!/bin/bash

read -p "Enter string: " string

reversed_string=$(echo "$string" | rev)
echo "Reversed string: $reversed_string"

```

```

$ bash reverse_string_10.sh
Enter string: OmKadam
Reversed string: madaKmO

```

11. Write a shell script to reverse the word order in a list of strings. For example, if the input is Hello World, output should be World Hello.

```
#!/bin/bash

read -p "Enter the strings separated by spaces: " input_string
reversed_string=$(echo "$input_string" | awk '{ for (i=NF; i>0; i--)
    printf "%s ", $i }')
echo "Reversed order word: $reversed_string"
```

```
$ bash word_order_11.sh
Enter the strings separated by spaces: Om Kadam
Reversed order word: Kadam Om
```

12. Write a shell script using case to either create a new file or delete an existing file.

bash script.sh --create newfile.txt should create this new file and bash script.sh --delete newfile.txt should delete this existing file.

Display "Not a valid argument" if neither --create nor --delete is specified. Hint: Use case \$1 to determine the argument option.

```
#!/bin/bash

if [ "$#" -lt 2 ]; then
    echo "Not a valid argument"
    exit 1
fi

case $1 in
    "--create")
        touch "$2"
        echo "File $2 created successfully"
        ;;
    "--delete")
        if [ -e "$2" ]; then
            rm "$2"
            echo "File $2 deleted successfully"
        else
            echo "File $2 does not exist"
        fi
        ;;
    *)
        echo "Not a valid argument"
        ;;
esac
```

~/A41 🕒 22:30:20
\$ **bash** file_12.sh
Not a valid argument

~/A41 🕒 22:30:21
\$ **bash** file_12.sh --create om.txt
File om.txt created successfully

~/A41 🕒 22:30:30
\$ **bash** file_12.sh --delete om.txt
File om.txt deleted successfully

13. Write a shell script for the following: accept a city name from user; using case statements, determine the country of this city. Provide multiple city names in each case statement.

Example of one case could be "Mumbai" | "Delhi" | "Pune") echo "The country is India" ;;
If options do not match, write one case statement as *) echo "To be added soon!".

```
#!/bin/bash

read -p "Enter a city: " city

case $city in
    "Mumbai" | "Delhi" | "Pune")
        echo "The country is India"
        ;;
    "Tokyo" | "Kobe" | "Kyoto")
        echo "The country is Japan"
        ;;
    *)
        echo "To be added soon!"
        ;;
esac
```

```
~/A41 🕒 22:33:23
$ bash city_country_13.sh
Enter a city: Mumbai
The country is India

~/A41 🕒 22:33:29
$ bash city_country_13.sh
Enter a city: Tokyo
The country is Japan
```

14. Write a shell script to convert user-given temperature in Celsius to Fahrenheit using bash calculator.

```
#!/bin/bash

read -p "Enter temperature in Celcius: " celsius

farheneit=$(echo "scale=2; ($celsius *9/5) + 32" | bc)

echo "$celsius in Farheneit is $farheneit."
```

```
~/A41 🕒 22:37:19
$ bash tempC_to_tempF.sh
Enter temperature in Celcius: 70
70 in Farheneit is 158.00.
```

15. Write a shell script to take two numbers from the user and choose arithmetic operations, i.e. add, subtract, multiply, divide, exponentiation, and return the corresponding result. Use case statements and bash calculator.

```
~/A41 🕒 22:46:38
$ bash operations_15.sh
Enter the first and second number43 33
Choose an arithmetic operation:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exponentiation
1
Addition result: 76
```

```

#!/bin/bash

read -p "Enter the first and second number" num1 num2

echo "Choose an arithmetic operation:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exponentiation"
read choice

case $choice in
    1)
        result=$(echo "$num1 + $num2" | bc)
        operation="Addition"
        ;;
    2)
        result=$(echo "$num1 - $num2" | bc)
        operation="Subtraction"
        ;;
    3)
        result=$(echo "$num1 * $num2" | bc)
        operation="Multiplication"
        ;;
    4)
        result=$(echo "$num1 / $num2" | bc)
        operation="Division"
        ;;
    5)
        result=$(echo "$num1 ^ $num2" | bc)
        operation="Exponentiation"
        ;;
    *)
        echo "Invalid choice."
        exit 1
        ;;
esac

echo "$operation result: $result"

```