

Task 01

part 1.

Create a Bash script which will take 3 numbers as command line arguments. It will print to the screen the larger of the three numbers.

Solution:

In this bash script code, **Echo** command is use to print string on console and **read** command is use to take input from console .

If command is a conditional Statement its syntax is . if [**var1** compare **var2**] .

\$ sign is used for compare two values and **-gt -ge** is operator use to compare values than.

Inside the square brackets n1 is compared with n2. Similarly **-ge** is used for greater than or Equal. after that use **then** keyword to continue it write echo command to show the output on terminal. And at last fi statement is used close the if condition like we close the loop with bracket.

```
#!/bin/bash
echo "Enter no 1"
read n1
echo "Enter no 2"
read n2
echo "Enter no 3"
read n3
if [ $n1 -ge $n2 ] && [ $n1 -ge $n3 ]; then
    echo "$n1 is the largest number."
elif [ $n2 -ge $n1 ] && [ $n2 -ge $n3 ]; then
    echo "$n2 is the largest number."
else
    echo "$n3 is the largest number."
fi
```

Code:

```
1 #!/bin/bash
2
3 echo "Enter no 1"
4 read n1
5 echo "Enter no 2"
6 read n2
7 echo "Enter no 3"
8 read n3
9
10 if [ $n1 -ge $n2 ] && [ $n1 -ge $n3 ]; then
11     echo "$n1 is the largest number."
12 elif [ $n2 -ge $n1 ] && [ $n2 -ge $n3 ]; then
13     echo "$n2 is the largest number."
14 else
15     echo "$n3 is the largest number."
16 fi
```

Output:

```
amadirfan@ubuntu:~/Desktop/lab 5$ ./task1.sh
Enter no 1
12
Enter no 2
14
Enter no 3
16
16 is the largest number.
amadirfan@ubuntu:~/Desktop/lab 5$
```

Part 2:

Create a Bash script which will print a message based upon which day of the week it is (eg. 'Happy day' for Wednesday, 'blessed' for Friday etc) using switch statement.

Solution

In variable day we take a day from our system using **\$(date +%A)**. date is carried out from our system. **Switch** is a statement like **if statement** in which we check cases between different cases. The one that matches output in that condition print on console.

```
#!/bin/bash

day=$(date +%A)

case $day in
    Monday)
        echo "focused.DBS day "
        ;;
    Tuesday)
        echo "motivation.DBS Lab"
        ;;
    Wednesday)
        echo "hoilyday!"
        ;;
    Thursday)
        echo "keep pushing. Busy day"
        ;;
    Friday)
        echo "Blessed, last day"
        ;;
    Saturday)
        echo "Relax and recharge."
        ;;
    Sunday)
        echo "ops.Last day of the week, "
        ;;
esac
```

Code:

```

1 #!/bin/bash
2
3 day=$(date +%A)
4
5 # Use a switch statement to print a message based on the day of the week
6 case $day in
7     Monday)
8         echo "focused.DBS day "
9         ;;
10    Tuesday)
11        echo "motivation.DBS Lab"
12        ;;
13    Wednesday)
14        echo "hollyday!"
15        ;;
16    Thursday)
17        echo "keep pushing. Busy day"
18        ;;
19    Friday)
20        echo "Blessed, last day"
21        ;;
22    Saturday)
23        echo "Relax and recharge."
24        ;;
25    Sunday)
26        echo "ops.Last day of the week, "
27        ;;
28 esac
29

```

Output:

```

amadirfan@ubuntu:~/Desktop/lab $ ./task2.sh
focused.DBS day
amadirfan@ubuntu:~/Desktop/lab $ s

```

Task2:

part1

Create a simple script which will print the numbers 1 - 10 (each on a separate line) and whether they are even or odd.

Solution

In this problem we use a counter variable in which we assign a value of **S1** which we provide when we run code. And we use a **until** keyword same for loop inside the bracket write the condition that if **counter variable greater than 10**. It will check 10 times inside the until statement we use do statement. do means do a task below that line in which 1st condition it checks if a number is divisible of 2 then it is an even number otherwise run the else statement in which print a message that the number is odd and close the if condition then increment in the counter variable by writing in double circle brackets counter++ and at last write done for complete the loop.

```
#!/bin/bash
counter=$1
until [ $counter -gt 10 ]
do
    if (( $counter % 2 == 0 ))
    then
        echo "$counter is even"
    else
        echo "$counter is odd"
    fi
    ((counter++))
done
```

Code:

```
1#!/bin/bash
2counter=$1
3until [ $counter -gt 10 ]
4do
5    if (( $counter % 2 == 0 ))
6    then
7        echo "$counter is even"
8    else
9        echo "$counter is odd"
10    fi
11    ((counter++))
12done
13
```

Output:

```
amadirfan@ubuntu:~/Desktop/lab 5$ ./task3.sh 1
1 is odd
2 is even
3 is odd
4 is even
5 is odd
6 is even
7 is odd
8 is even
9 is odd
10 is even
amadirfan@ubuntu:~/Desktop/lab 5$
```

part2

Write a program that read number as input, calculate, and return the summation of the all digits of the number.

Example:

If given number: 745

Then result will be: $(7+4+5) = 16$

Solution

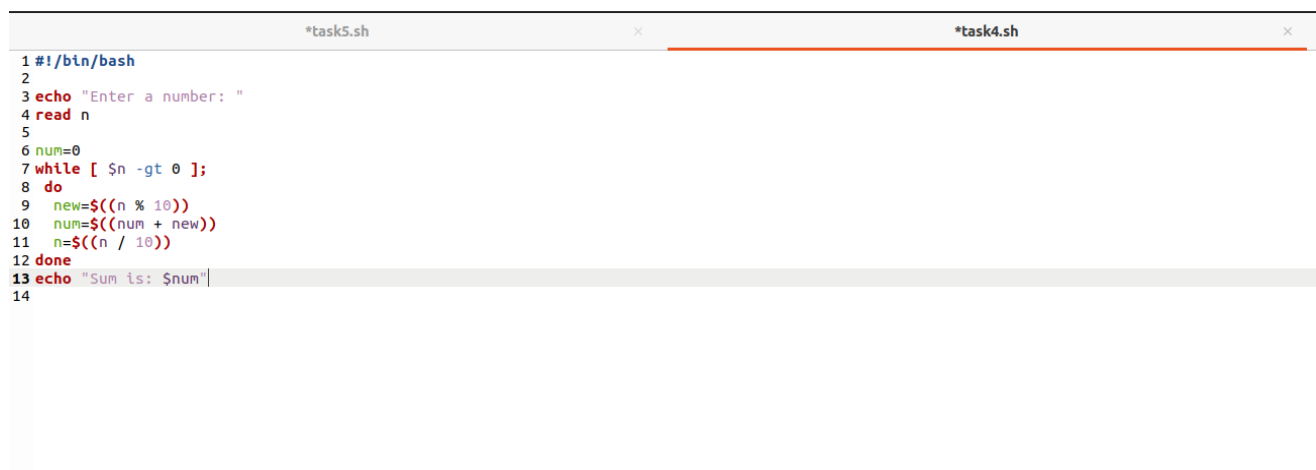
In this problem we get a number from terminal and create a variable **num** and assign it a value of 0 at the time of initiation the we use the while loop and the condition in that is when the number from the terminal is greater than 0 take the mod of that number and assign it into a new variable and sum it to the variable this process runs until the condition of greater than 0 false and the variable n is updated and at last of while write done to complete the loop.

```
#!/bin/bash

echo "Enter a number: "
read n

num=0
while [ $n -gt 0 ];
do
    new=$((n % 10))
    num=$((num + new))
    n=$((n / 10))
done
echo "Sum is: $num"
```

Code:



```
*task5.sh  ×  *task4.sh  ×
1 #!/bin/bash
2
3 echo "Enter a number: "
4 read n
5
6 num=0
7 while [ $n -gt 0 ];
8 do
9     new=$((n % 10))
10    num=$((num + new))
11    n=$((n / 10))
12 done
13 echo "Sum is: $num"
14
```

Output:



```
amadirfan@ubuntu:~/Desktop/lab $ ./task4.sh
Enter a number:
123
Sum is: 6
amadirfan@ubuntu:~/Desktop/lab $
```

Task 3:

part1

Write a shell script which takes a positive integer as an argument on the terminal and then checks if it is a palindrome or not. In order to find the reverse of this number it must be passed to function

named `reverse()`, which computes the reverse and passes both, the number and its reverse to another function called `palindromeCheck()`. `palindromeCheck()` then compares the numbers and echoes if the number is a palindrome or not.

Solution

In this problem we create two function one is named for **palindrome** and **reverse** .

Palindrome number. It is a number when it reversed it remain same as the number.

In that we created a reverse function which takes the mod of number and another number multiplied by 10 for example if it is 66 we take mod of it 6 is carried out then multiply the number with 10 it is 690 then add remaining number which is 6 and it is 66. after creating reverse function Now create **palindromecheck()** in which number is checked with the number that was reversed if the value is same as reverse then it is a palindrome number.

```
#!/bin/bash

if [ $# -ne 1 ]; then
    echo "Error: Please provide exactly 1 positive integer as argument."
    exit 1
fi

number=$1

reverse() {
    reverse=0
    local num=$1
    while [ $num -gt 0 ]; do
        reverse=$((reverse * 10 + num % 10))
        num=$((num / 10))
    done
    echo $reverse
}

palindromeCheck() {
    local num=$1
    local rev=$2
    if [ $num -eq $rev ]; then
        echo "$num is a palindrome."
    else
        echo "$num is not a palindrome."
    fi
}

rev=$((reverse $number))

palindromeCheck $number $rev
```

Code:

```
task5.sh x task6.sh x
1#!/bin/bash
2
3if [ $# -ne 1 ]; then
4    echo "Error: Please provide exactly 1 positive integer as argument."
5    exit 1
6fi
7
8number=$1
9
10reverse() {
11    reverse=0
12    local num=$1
13    while [ $num -gt 0 ]; do
14        reverse=$((reverse * 10 + num % 10))
15        num=$((num / 10))
16    done
17    echo $reverse
18}
19
20palindromeCheck() {
21    local num=$1
22    local rev=$2
23    if [ $num -eq $rev ]; then
24        echo "$num is a palindrome."
25    else
26        echo "$num is not a palindrome."
27    fi
28}
29
30rev=$(reverse $number)
31
32palindromeCheck $number $rev
sh Tab Width: 8 Ln 1, Col 1 INS
```

Output:

```
amadirfan@ubuntu:~/Desktop/lab $ ./task5.sh 121
121 is a palindrome.
amadirfan@ubuntu:~/Desktop/lab $
```

Part 2:

Write a shell script which reads a number from the user and passes it to function named **factorial()**. This function finds the factorial of the number and prints it on the terminal. This task must be done using recursion.

Solution

In this problem we have to find factorial of a number using recursion we created a function named as **factorial** in which we created a local number and assigns it a value of **\$1** and use recursion that if the number is 0 then print 1 other wise go to else statement where in this step the function calls itself again and again which is called recursion until the answer is 1. Then add up all the result and at last we get a number from terminal and give that number to the factorial function which returns its result and we save.

```
#!/bin/bash

echo "Enter a number: "
read a

factorial() {
    local number=$1
    if [ $number -eq 0 ]; then
        echo 1
    else
        echo $((number * $(factorial $((number - 1)))))
    fi
}
```

```
}

factorial=$(factorial $a)

echo "The factorial of $a is $factorial."
```

Code:

```
1 #!/bin/bash
2
3 echo "Enter a number: "
4 read a
5
6 factorial() {
7     local number=$1
8     if [ $number -eq 0 ]; then
9         echo 1
10    else
11        echo $((number * $(factorial $((number - 1)))))
12    fi
13 }
14
15 factorial=$(factorial $a)
16
17 echo "The factorial of $a is $factorial."
18
```

Output:

```
amadirfan@ubuntu:~/Desktop/lab 5$ ./task6.sh
Enter a number:
5
The factorial of 5 is 120.
amadirfan@ubuntu:~/Desktop/lab 5$
```