/*IF*/

1. WAP to check whether the product of two given numbers is odd or even.

```
#!/bin/bash

<<Doc
Name:Jagadeesh
Date:
Description:
Sample i/p:
Sample o/p:
Doc

echo "ENTER TWO NUMBERS"
read num1 num2
prod=`expr $num1 \* $num2`
echo "$prod"

if [ $(($prod % 2)) -eq 0 ]
then
        echo "product is even"
else
        echo "product is odd"
fi
```

---

2. WAP to check whether the entered number is multiple of 5.

```
#!/bin/bash

<<Doc
Name:Jagadeesh
Date:14-09-2022
Description:check whether the entered number multiple of 5
Sample i/p:Any number
Sample o/p:is entered number Multiple by 5 / not
Doc

echo "Enter number"
read num1
```

```
if [ $((num1 % 5)) -eq 0 ]
then
        echo "Number is multiple of 5"
else
        echo "Number is not multiple of 5"
fi
```

---

3. WAP, given three sides of a triangle, to check whether the triangle is perfect right triangle.

```
#!/bin/bash

<<Doc
Name:Jagadeesh
Date:14-09-2022
Description:given 3 sides of triangle,to check the triangle is perfect
           right triangle?
Sample i/p:Triangle a b c values
Sample o/p:Triangle is right angle \ not
Doc

echo "Enter values of Triangle sides a b c"
read small1
read small2
read large

if [ $(( $small1 ** 2) + ( $small2 ** 2))) -eq $(( $large ** 2)) ]
then
    echo " Triangle is right angled "
else
    echo " Triangle is not right angled"
fi
```

---

4. WAP to check whether given year is leap year or not.

```
#!/bin/bash

<<Doc
```

```
Name:Jagadeesh
Date:14-09-2022
Description:check whether given number is leap year or not.
Sample i/p:
Sample o/p:
Doc

read year
if [ $((year % 100)) -eq 0 ]
then
    if [ $((year % 400)) -eq 0 ]
    then
        echo Leap year
    else
        echo not a leap year
    fi
elif [$((year % 4)) -eq 0]
then
        echo Leap year
else
        echo Not a leap year
fi

-------------------------------------------------**************************
*****************************----------------------------------------------
--
```

5. WAP that reads in 2 integers and determines and prints if the first is a multiple of the second.

```
#!/bin/bash

<<Doc
Name:Jagadeesh
Date:14-09-2022
Description:if the first is multiple of the second
Sample i/p:
Sample o/p:
Doc

echo "Enter the First integer numbers"
read num1
echo "Enter the Second integer numbers"
read num2
```

```
if [ $(($num1 % $num2)) -eq 0 ]
then
echo " First number is Multiple of Second Number "
else
echo " First number is not Multiple of Second Number "
fi

-------------------------------------------------**************************
*****************************-------------------------------------------
--

/*FOR, WHILE */


-------------------------------------------------**************************
*****************************-------------------------------------------
--


6. WAP to find the smallest and largest of entered n numbers

#!/bin/bash

<<Doc
Name:Jagadeesh
Date:14-09-2022
Description:find the smallest and largest of entered n number.
Sample i/p:
Sample o/p:
Doc

read n
read num
small=$num
large=$num
for i in `seq $((n-1))`
do
    read num
    if [ $num -lt $small ]
    then
        small=$num
    elif [ $num -gt $large ]
    then
        large=$num
        fi
    done
```

```
echo "small=$small"
echo "large=$large"

-------------------------------------------------**************************
****************************-----------------------------------------
--
```

7. WAP to print all the even numbers from 1 to 100.

```
#!/bin/bash

<<Doc
Name:Jagadeesh
Date:15-09-2022
Description:to print all the even numbers from 1 to 100
Sample i/p:
Sample o/p:
Doc

read -p "Enter the number1: " M
read -p "Enter the number2: " N
#M=$((M+1))
#N=$((N-1))
while [ $M -le $N ]
do
    if [ `expr $M % 2` -eq 0 ]
then
    echo "$M"
fi
M=`expr $M + 1`
done
```

```
-------------------------------------------------**************************
****************************-----------------------------------------
--
```

8. WAP to print the ‘n’ Fibonacci terms.

```
#!/bin/bash

<<Doc
Name:Jagadeesh
```

```
Date:
Description:
Sample i/p:
Sample o/p:
Doc

echo "Enter the number"
read n
num1=0
num2=1
count=2
echo " numbers upto $n terns "
echo "$num1"
echo "$num2"
while [ $count -lt $n ]
do
    count=`expr $count + 1`
    a=`expr $num1 + $num2`
    echo "$a"
    num1=$num2
    num2=$a
done


--------------------------------------------------**************************
*****************************-----------------------------------------------
--
```

9. WAP to check whether a number is palindrome or not.

```
#!/bin/bash

<<Doc
Name:Jagadeesh
Date:
Description:
Sample i/p:
Sample o/p:
Doc

echo "Enter number"
read num
```

```
for i in $num
do
   a=`expr $num % 10`
   b=`expr $a \* 10`
   c=`expr $b + 2`
   d=`expr $c \* 10`
   e=`expr $d + 1`
done

if [ $e = $num ]
then
    echo " number is palindrone "
else
    echo " number is not palindrone "
fi
```

---

10. WAP to check whether a given number is prime or not.

```
#!/bin/bash
<<Doc
Name:Jagadeesh
Date:
Description:
Sample i/p:
Sample o/p:
Doc

echo " Enter the Number "
read n
count=0
while [ $i -le `expr $number / 2` ]
do
    if [ `expr $number % $i` -eq 0 ]
    then
        count=1
    fi
    i=$((i+1))
done
```

```
if [ $count -eq 1 ]
then
    echo "$number is not a prime number"
else
    echo "$number is a prime number"
fi
```

11. WAP to print the pattern as follows if input is 5:

```
#!/bin/bash

<<Doc
Name:Jagadeesh
Date:
Description:
Sample i/p:
Sample o/p:
Doc

echo " Enter the number "
read n
for i in `seq $n`
do
    for j in `seq $((${n} - ${i}))`
    do
        echo -n " "
    done
    for k in `seq $-i`
    do
        echo -n "*"
    done
    echo
done
```