1. **Leap year and checking odd or even number**

```
echo "Enter the  value of year"
read year
if [ ` expr $year % 4 ` -eq 0 ]
then
echo "It is a leap year"
else
echo "It is not a leap year"
fi
echo Enter the number
read num
if [ ` expr $num % 2 ` -eq 0 ]
then
echo "It is an even number"
else
echo "It is an odd number"
fi
```

Output :

```
Enter the  value of year
2000
It is a leap year
Enter the number
34
It is an even number
```

2. **Leap year with 3 conditions**

```
echo Enter the value of a
read a
echo Enter the value of b
read b
echo Enter the value of c
read c
if [ $a -gt $b -a $a -gt $c ]
then
echo $a is greater then $b and $c
elif [ $b -gt $a -a $b -gt $c ]
```

```
then
echo $b is greater then $a and $c
else
echo $c is greater then $a and $b
```

Output :

```
Enter the value of a
5
Enter the value of b
4
Enter the value of c
3
5 is greater then 4 and 3
```

3. **Multiplication table & printing the natural numbers**

```
echo Enter the number for multiplication
read n
echo Enter the the starting point
read i
while [ $i -le 10 ]
do
        echo " $n * $i = `expr $n \* $i` "
        i=`expr $i + 1`
done
echo To display the natural number
echo Enter the value
read n
echo Enter the value of i
read i
while [ $i -le $n ]
do
        echo the value of i is $i
        i=`expr $i + 1`
done
```

Output:

```
Enter the number for multiplication
```

```
2
Enter the the starting point
1
 2 * 1 = 2
 2 * 2 = 4
 2 * 3 = 6
 2 * 4 = 8
 2 * 5 = 10
 2 * 6 = 12
 2 * 7 = 14
 2 * 8 = 16
 2 * 9 = 18
 2 * 10 = 20
To display the natural number
Enter the value
10
Enter the value of i
1
the value of i is 1
the value of i is 2
the value of i is 3
the value of i is 4
the value of i is 5
the value of i is 6
the value of i is 7
the value of i is 8
the value of i is 9
the value of i is 10
```

4. LCM and GCD

```
echo Enter the two numbers M and N
read m
read n
temp=` expr $m \* $n `
echo "The value of temp is $temp"
while [ $m -ne $n ]
do
if [ $m -gt $n ]
```

```
then
m=` expr $m - $n `
else
n=` expr $n - $m `
fi
done
gcd=$n
LCM=`expr $temp / $gcd`
echo LCM = $LCM
echo GCD = $gcd
```

Output:

```
Enter the two numbers M and N
3
2
The value of temp is 6
LCM = 6
GCD = 1
```

5. **Break and continue statement**

```
echo Example for break statement
echo Enter the number
read a
while [ $a -lt 10 ]
do
echo $a
if [ $a -eq 5 ]
then
break
fi
a=`expr $a + 1`
done
```

Output:

```
Example for break statement
Enter the number
```

```
1
1
2
3
4
5
```

6. **Reverse order of a string**

```
echo "Enter the string to print in reverse order"
read str
len=`echo $str | wc -c`
echo $len
len=`expr $len - 1`
echo "Length of the string $len"
echo "Original string is $str"
echo "Length of string is : $len"
echo "Reversed string is : "
while [ $len -ge 1 ]
do
rev=`echo $str | cut -c $len`
echo $rev
len=`expr $len - 1`
done
```

Output:

```
Enter the string to print in reverse order
reva
5
Length of the string 4
Original string is reva
Length of string is : 4
Reversed string is :
a
v
e
r
```

7. **Calculate the electricity bill**

```
echo "Enter the units consumed"
echo "Enter the units"
read units
if [ $units -gt 10 -o $units -lt 15 ]
then
net=`expr $units \* 2 + 25`
echo "The value of units is $net"
elif [ $units -gt 40 ]
then
net1=`expr $units \* 3 + 25`
echo "The value of units is $net1"
else
echo "Units are crossed"
fi
```

Output:

```
Enter the units consumed
Enter the units
11
The value of units is 47
```

8. **Palindrome**

```
echo Enter a string
read str
size=`echo $str | wc -c`
len=`expr $size - 1`
k=`expr $len / 2`
i=1
while [ $i -le $k ]
do
first=`echo $str | cut -c $i`
ch=`expr $len - $i + 1`
last=`echo $str | cut -c $ch`
if [ $first != $last ]
```

```
then
echo $str is not a palindrome
exit
fi
i=`expr $i + 1`
done
echo $str is a palindrome
```

Output:

```
Enter a string
mom
mom is a palindrome

Enter a string
reva
reva is not a palindrome
```

9. Frequency of occurance of a character

```
echo Enter a string
cat > str
echo Enter the character to be searched
read char
i=1
count=0
len=`cat str | wc -c`
echo The length of the string is $len
while [ $i -lt $len ]
do
        ch=`cat str | cut -c $i`
        if [ `expr $ch` -eq $char ]
        then
                count=`expr $count + 1`
        fi
        i=`expr $i + 1`
done
echo The given string is `cat str`
echo $count times
```

Output:

```
Enter the character to be searched
5
The length of the string is 8
The given string is 2345456
2 times
```

10. Prime number in given range

```
echo Enter the start number
read m
echo Enter the end number
read n
echo "The prime numbers for given range ($m to $n)"
while [ $m -le $n ]
do
i=2
flag=1
while [ $i -lt $m ]
do
if [ `expr $m % $i` -eq 0 ]
then
flag=0
break
else
i=`expr $i + 1`
fi
done
if [ $flag -eq 1 ]
then
echo $m
fi
m=`expr $m + 1`
done
```

Output:

```
Enter the start number
1
```

```
Enter the end number
100
The prime numbers for given range (1 to 100)
1
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
83
89
97
```

11. Fabanocci Series

```
f1=1
f2=1
echo how many numbers
read n
f3=`expr $f1 + $f2`
```

```
i=3
echo $f1
echo $f2
while [ $i -le $n ]
do
echo $f3
f1=$f2
f2=$f3
f3=`expr $f1 + $f2`
i=`expr $i + 1`
done
```

Output:

```
how many numbers
10

1
1
2
3
5
8
13
21
34
55
```

12. Student marks, sum and average calculation

```
a=0
while [ $a -lt 10 ]
do
        b=$a
        while [ $b -gt 0 ]
        do
                echo -n $b
                b=`expr $b - 1`
        done
```

```
        echo $b
        a=`expr $a + 1`
done
echo Enter the 5 subject marks of the student
read m1
read m2
read m3
read m4
read m5
sum=`expr $m1 + $m2 + $m3 + $m4 + $m5`
echo The sum of the marks is $sum
avg=`expr $sum / 5`
echo The value of average is $avg
if [ $m2 -lt 40 -o $m3 -lt 40 -o $m4 -lt 40 ]
then
echo FAIL
elif [ $avg -ge 70 ]
then
echo Distinction
elif [ $avg -ge 60 ]
then
echo First class
elif [ $avg -ge 50 ]
then
echo Second class
elif [ $avg -ge 40 ]
then
echo Third class
else
echo Not passes
fi
```

Output:

```
0
10
210
3210
43210
543210
```

```
6543210
76543210
876543210
9876543210
Enter the 5 subject marks of the student
34
67
56
78
89
The sum of the marks is 324
The value of average is 64
First class
```

13. Arithmetic operations

```
echo Enter the value of a
read a
echo Enter the value of b
read b
c=`echo $a + $b | bc`
d=`echo $a - $b | bc`
e=`echo $a \* $b | bc`
f=`echo $a / $b | bc`
g=`echo $a % $b | bc`
echo Addition of two numbers is $c
echo Subtraction of two numbers is $d
echo Multiplication of two numbers is $e
echo Division of two numbers is $f
echo Modulus of two numbers is $g
echo "-----------------------------------"
echo Enter the value of c
read c
echo Enter the value of d
read d
res=`echo $c + $d | bc`
res1=`echo $c - $d | bc`
res2=`echo $c / $d | bc`
res3=`echo $c % $d | bc`
```

```
echo a + b = $res
echo a - b = $res1
echo a / b = $res2
echo a % b = $res3
```

Output:

```
Enter the value of a
5
Enter the value of b
3
Addition of two numbers is 8
Subtraction of two numbers is 2
Multiplication of two numbers is 15
Division of two numbers is 1
Modulus of two numbers is 2
-----------------------------------
Enter the value of c
4
Enter the value of d
2
a + b = 6
a - b = 2
a / b = 2
a % b = 0
```

14. Menu driven and switch case Factorial of the number

```
echo 1. list of files
echo 2. current date
echo 3. process status
echo 4. logged in users
echo 5. present working directory
echo 6. quit
echo Enter your choice
read ch
```

```
case $ch in
1) ls -l;;
2) date;;
3) ps;;
4) who;;
5) pwd;;
6) exit;;
*) echo invalid choice;;
Esac
echo Factorial of a number
echo Enter the number
read num
fact=1
while [ $num -gt 1 ]
do
fact=$((fact * num))
num=$((num - 1))
done
echo Factorial of the above number is $fact
echo "----------------------------------------------"
```

Output:

```
1. list of files
2. current date
3. process status
4. logged in users
5. present working directory
6. quit
Enter your choice
2
Thursday 09 June 2022 09:49:41 PM IST
----------------------------------------------
Factorial of a number
Enter the number
6
Factorial of the above number is 720
```

---

15. Vowels in the String

```
echo Enter the line of text
read string
num=$(echo $string | grep -o "[0-9]" | wc --lines)
vow=$(echo $string | grep -o -i "[aeiou]" | wc --lines)
con=$(echo $string | grep -o -i "[bcdfghjklmnpqrstvwxyz]" | wc --lines)
vowcon=$(echo $string | grep -o -i "[abcdefghijklmnopqrstuvwxyz]" | wc --lines)
echo The given string has $num numbers
echo The given string has both combined $vowcon
echo The given string has $vow vowels
echo The given string has $con consonents
echo $count
echo $i
```

Output:

```
Enter the line of text
reva university
The given string has 0 numbers
The given string has both combined 14
The given string has 6 vowels
The given string has 8 consonents
```

16. Checking the file permission

```
echo Enter the filename
read fn
if [ -r $fn -a -x $fn ]
then
echo "The file has only read and execute permission"
else
echo "The file has invalid permission"
fi
```

Output:

```
$ chmod 555 stdlist
-r-xr-xr-x. 1 balajiraod balajiraod 119 Apr 29 14:14 stdlist
```

Now run the program

```
$ sh program16.sh
```

Enter the filename
stdlist
The file has only read and execute permission

17. Find the sum of digits

```
sum=0
echo "Enter the number"
read num
n=$sum
while [ $num -gt 0 ]
do
rem=`expr $num % 10`
sum=`expr $sum + $rem`
num=`expr $num / 10`
done
echo The sum of digits is $sum
```

Output:

Enter the number
12345
The sum of digits is 15

18. List the files in current directory

```
echo Enter the directory name
read dir
if [ -d $dir ]
then
echo "List of files in the directory"
ls -l $dir | egrep '^d'
else
echo "Enter proper directory name"
fi
```

Output:

Enter the directory name
commands
List of files in the directory

```
drwxr-xr-x. 1 balajiraod balajiraod 0 Jun  4 18:22 hello.
drwxr-xr-x. 1 balajiraod balajiraod 0 Jun  4 18:22 Reva University
```

19. Create a file and compress it

```
echo "Enter the file name to be compressed"
read fname
wc -c $fname
gzip $fname
gzip -l $fname.gz
echo "Decompressing the file"
gunzip $fname.gz
wc -c $fname
count=0
```

Output:

```
Enter the file name to be compressed
stdlist
119 stdlist
        compressed       uncompressed  ratio uncompressed_name
             127              119  15.1% stdlist
Decompressing the file
119 stdlist
```

20. Program to convert the content of a given test from lowercase to uppercase and also count the number of character, lines, words of the result and text

```
fn="abcdefghijklmnopqrstuvwxyz"
for i in $fn
do
i=`echo $i | tr '[a-z]' '[A-Z]'`
echo $i
done
len=`expr length $fn`
echo The length of the string is $len
```

Output: ABCDEFGHIJKLMNOPQRSTUVWXYZ
The length of the string is 26