1. Leap Year Check (with blinking output using tput)

```
#!/bin/bash
echo "Enter a year:"
read year

if [ $((year % 400)) -eq 0 ] || { [ $((year % 4)) -eq 0 ] && [ $((year % 100)) -ne 0 ]; }
then
    tput blink
    echo "$year is a Leap Year"
else
    tput blink
    echo "$year is NOT a Leap Year"
fi
tput sgr0
```

2. File Modified Check (using diff)

```
#!/bin/bash
echo "Enter the file name:"
read file

cp $file backup.txt
echo "Modify the file if you want, then press Enter..."
read

diff $file backup.txt > /dev/null
if [ $? -eq 0 ]; then
    echo "File is NOT modified"
else
    echo "File is MODIFIED"
fi
```

3. Fibonacci Series up to N

```
#!/bin/bash
echo "Enter a number:"
read n

a=0
b=1

echo "Fibonacci series up to $n:"
while [ $a -le $n ]
do
    echo -n "$a "
    fn=$((a + b))
    a=$b
    b=$fn
done
echo
```

4. Calculator with Functions (Interactive)

```
#!/bin/bash

add() { echo "Result: $(($1 + $2))"; }
sub() { echo "Result: $(($1 - $2))"; }
mul() { echo "Result: $(($1 * $2))"; }
div() { echo "Result: $(($1 / $2))"; }

echo "Enter two numbers:"
read a b

echo "Enter operation (add/sub/mul/div):"
read op

case $op in
    add) add $a $b ;;
    sub) sub $a $b ;;
    mul) mul $a $b ;;
    div) div $a $b ;;
    *) echo "Invalid operation" ;;
esac
```

5. Calculator (Command Line Args)

```
#!/bin/bash

add() { echo "Result: $(($1 + $2))"; }
sub() { echo "Result: $(($1 - $2))"; }
mul() { echo "Result: $(($1 * $2))"; }
div() { echo "Result: $(($1 / $2))"; }

op=$1
a=$2
b=$3

case $op in
    add) add $a $b ;;
    sub) sub $a $b ;;
    mul) mul $a $b ;;
    div) div $a $b ;;
    *) echo "Usage: $0 {add|sub|mul|div} num1 num2" ;;
esac
```

6. Odd/Even Check (Red prompt, Blue blinking output)

```
#!/bin/bash

echo -e "\e[31mEnter a number:\e[0m"
read num

if [ $((num % 2)) -eq 0 ]; then
    tput setaf 4; tput blink
    echo "$num is EVEN"
else
    tput setaf 4; tput blink
```

```
    echo "$num is ODD"
fi
tput sgr0
```

7. Square and Square Root

```
#!/bin/bash
echo "Enter a number:"
read num

square=$((num * num))
sqrt=$(echo "scale=2; sqrt($num)" | bc)

echo "Square: $square"
echo "Square Root: $sqrt"
```