# 1. Largest of Three Numbers

#!/bin/bash  
echo "Enter three numbers:"  
read a b c  
if [ $a -ge $b ] && [ $a -ge $c ]; then  
 echo "$a is the largest"  
elif [ $b -ge $a ] && [ $b -ge $c ]; then  
 echo "$b is the largest"  
else  
 echo "$c is the largest"  
fi

# 2. Check Leap Year

#!/bin/bash  
echo "Enter a year:"  
read year  
if (( (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0) )); then  
 echo "$year is a leap year"  
else  
 echo "$year is not a leap year"  
fi

# 3. Valid Triangle Check

#!/bin/bash  
echo "Enter three angles of a triangle:"  
read a b c  
sum=$((a + b + c))  
if [ $sum -eq 180 ]; then  
 echo "Valid triangle"  
else  
 echo "Invalid triangle"  
fi

# 4. Character Type Check

#!/bin/bash  
echo "Enter a character:"  
read ch  
if [[ $ch =~ [A-Za-z] ]]; then  
 echo "Alphabet"  
elif [[ $ch =~ [0-9] ]]; then  
 echo "Digit"  
else  
 echo "Special Character"  
fi

# 5. Calculate Profit or Loss

#!/bin/bash  
echo "Enter Cost Price and Selling Price:"  
read cp sp  
if [ $sp -gt $cp ]; then  
 echo "Profit of $((sp - cp))"  
elif [ $cp -gt $sp ]; then  
 echo "Loss of $((cp - sp))"  
else  
 echo "No profit, no loss"  
fi

# 6. Print Even and Odd Numbers (1 to 10)

#!/bin/bash  
echo "Even Numbers:"  
for ((i=1; i<=10; i++)); do  
 if (( i % 2 == 0 )); then echo $i; fi  
done  
  
echo "Odd Numbers:"  
for ((i=1; i<=10; i++)); do  
 if (( i % 2 != 0 )); then echo $i; fi  
done

# 7. Print Multiplication Table

#!/bin/bash  
echo "Enter a number:"  
read num  
for ((i=1; i<=10; i++)); do  
 echo "$num x $i = $((num \* i))"  
done

# 8. Factorial of a Number

#!/bin/bash  
echo "Enter a number:"  
read n  
fact=1  
for ((i=1; i<=n; i++)); do  
 fact=$((fact \* i))  
done  
echo "Factorial of $n is $fact"

# 9. Sum of Even Numbers (1 to 10)

#!/bin/bash  
sum=0  
for ((i=2; i<=10; i+=2)); do  
 sum=$((sum + i))  
done  
echo "Sum of even numbers from 1 to 10 is $sum"

# 10. Sum of Digits

#!/bin/bash  
echo "Enter a number:"  
read num  
sum=0  
while [ $num -gt 0 ]; do  
 digit=$((num % 10))  
 sum=$((sum + digit))  
 num=$((num / 10))  
done  
echo "Sum of digits: $sum"

# 11. Basic Calculator

#!/bin/bash  
echo "Enter two numbers:"  
read a b  
echo "Enter operator (+, -, \*, /):"  
read op  
  
case $op in  
 +) echo "$a + $b = $((a + b))" ;;  
 -) echo "$a - $b = $((a - b))" ;;  
 \\*) echo "$a \* $b = $((a \* b))" ;;  
 /)   
 if [ $b -ne 0 ]; then  
 echo "$a / $b = $((a / b))"  
 else  
 echo "Division by zero not allowed"  
 fi ;;  
 \*) echo "Invalid operator" ;;  
esac

# 12. Print Days of the Week

#!/bin/bash  
days=("Sunday" "Monday" "Tuesday" "Wednesday" "Thursday" "Friday" "Saturday")  
for day in "${days[@]}"; do  
 echo $day  
done

# 13. First 4 Months with 31 Days

#!/bin/bash  
months=("January" "March" "May" "July")  
for month in "${months[@]}"; do  
 echo $month  
done

# 14. Functions

#!/bin/bash  
  
amstrong() {  
 echo "Enter a number:"  
 read num  
 n=$num  
 sum=0  
 while [ $n -gt 0 ]; do  
 r=$((n % 10))  
 sum=$((sum + r\*r\*r))  
 n=$((n / 10))  
 done  
 if [ $sum -eq $num ]; then  
 echo "$num is an Armstrong number"  
 else  
 echo "$num is not an Armstrong number"  
 fi  
}  
  
palindrome() {  
 echo "Enter a number:"  
 read num  
 n=$num  
 rev=0  
 while [ $n -gt 0 ]; do  
 r=$((n % 10))  
 rev=$((rev \* 10 + r))  
 n=$((n / 10))  
 done  
 if [ $rev -eq $num ]; then  
 echo "$num is a Palindrome"  
 else  
 echo "$num is not a Palindrome"  
 fi  
}  
  
fibonacci() {  
 echo "Enter number of terms:"  
 read n  
 a=0  
 b=1  
 for ((i=0; i<n; i++)); do  
 echo -n "$a "  
 fn=$((a + b))  
 a=$b  
 b=$fn  
 done  
 echo  
}  
  
prime() {  
 echo "Enter a number:"  
 read n  
 if [ $n -le 1 ]; then  
 echo "$n is not Prime"  
 return  
 fi  
 for ((i=2; i\*i<=n; i++)); do  
 if ((n % i == 0)); then  
 echo "$n is Composite"  
 return  
 fi  
 done  
 echo "$n is Prime"  
}  
  
decimal\_to\_binary() {  
 echo "Enter a decimal number:"  
 read n  
 binary=""  
 while [ $n -gt 0 ]; do  
 binary=$((n % 2))$binary  
 n=$((n / 2))  
 done  
 echo "Binary equivalent: $binary"  
}  
  
# Menu  
echo "Choose operation:"  
echo "1. Armstrong"  
echo "2. Palindrome"  
echo "3. Fibonacci"  
echo "4. Prime/Composite"  
echo "5. Decimal to Binary"  
read choice  
  
case $choice in  
 1) amstrong ;;  
 2) palindrome ;;  
 3) fibonacci ;;  
 4) prime ;;  
 5) decimal\_to\_binary ;;  
 \*) echo "Invalid option" ;;  
esac