1. Print Current Date and Time

```
echo "Current Date and Time: $(date '+%Y-%m-%d %H:%M:%S')"
```

2. File Existence Check

```
echo "Enter file name:"
read filename
if [ -e "$filename" ]; then
    echo "File exists"
else
    echo "File does not exist"
fi
```

3. Basic Calculator

```
echo "Enter first number:"
read num1
echo "Enter second number:"
read num2
echo "Choose operation (+ - * /):"
read op
case $op in
    +) result=$((num1 + num2)) ;;
    -) result=$((num1 - num2)) ;;
    \*) result=$((num1 * num2)) ;;
    /) result=$((num1 / num2)) ;;
    *) echo "Invalid operation" ;;
esac
echo "Result: $result"
```

4. Check for Even or Odd Number

```
echo "Enter a number:"
read num
if [ $((num % 2)) -eq 0 ]; then
    echo "$num is even"
else
    echo "$num is odd"
fi
```

5. Factorial of a Number

```
echo "Enter a number:"
read num
fact=1
```

```
for ((i = 1; i <= num; i++)); do
    fact=$((fact * i))
done
echo "Factorial of $num is $fact"
```

6. Command-line Argument Sum

```
if [ $# -ne 2 ]; then
    echo "Please provide two numbers as arguments."
    exit 1
fi
sum=$(( $1 + $2 ))
echo "Sum: $sum"
```

7. Count Number of Lines in a File

```
echo "Enter file name:"
read filename
if [ -e "$filename" ]; then
    lines=$(wc -l < "$filename")
    echo "Number of lines: $lines"
else
    echo "File does not exist"
fi
```

8. Display a List of Files in a Directory

```
echo "Enter directory name:"
read dirname
if [ -d "$dirname" ]; then
    echo "Files in $dirname:"
    ls "$dirname"
else
    echo "Directory does not exist"
fi
```

9. Create a Backup Script

```
echo "Enter source directory:"
read src
echo "Enter destination directory:"
read dest
if [ -d "$src" ]; then
    cp "$src"/*.txt "$dest"
    echo "Backup complete"
else
    echo "Source directory does not exist"
fi
```

10. User Input Validation

```
echo "Enter a number:"
read num
if ! [[ "$num" =~ ^[0-9]+$ ]]; then
    echo "Invalid input. Please enter a valid number."
else
    echo "You entered: $num"
fi
```

11. Print Multiplication Table

```
echo "Enter a number:"
read num
for i in {1..10}; do
    echo "$num * $i = $((num * i))"
done
```

12. Check If String is Palindrome

```
echo "Enter a string:"
read str
reverse=$(echo "$str" | rev)
if [ "$str" == "$reverse" ]; then
    echo "$str is a palindrome"
else
    echo "$str is not a palindrome"
fi
```

13. Check Disk Space

```
disk_usage=$(df / | grep / | awk '{ print $5 }' | sed 's/%//g')
if [ $disk_usage -gt 90 ]; then
    echo "Warning: Disk space usage is above 90%."
else
    echo "Disk space usage is at $disk_usage%."
fi
```

14. Find the Largest of Three Numbers

```
echo "Enter three numbers:"
read num1 num2 num3
if [ "$num1" -ge "$num2" ] && [ "$num1" -ge "$num3" ]; then
    echo "Largest number: $num1"
elif [ "$num2" -ge "$num1" ] && [ "$num2" -ge "$num3" ]; then
    echo "Largest number: $num2"
else
    echo "Largest number: $num3"
fi
```

15. Reverse a String

```
echo "Enter a string:"
read str
reverse=$(echo "$str" | rev)
echo "Reversed string: $reverse"
```

16. Create a Directory if It Does Not Exist

```
echo "Enter directory name:"
read dirname
if [ -d "$dirname" ]; then
    echo "Directory already exists."
else
    mkdir "$dirname"
    echo "Directory created."
fi
```

17. Simple Menu-driven Program

```
while true; do
    echo "Choose an option:"
    echo "1. Display date"
    echo "2. List files"
    echo "3. Exit"
    read choice
    case $choice in
        1) echo "Current Date and Time: $(date)" ;;
        2) ls ;;
        3) exit 0 ;;
        *) echo "Invalid choice" ;;
    esac
done
```

18. Check Prime Number

```
echo "Enter a number:"
read num
is_prime=1
for ((i = 2; i <= num / 2; i++)); do
    if [ $((num % i)) -eq 0 ]; then
        is_prime=0
        break
    fi
done
if [ $is_prime -eq 1 ]; then
    echo "$num is a prime number"
else
    echo "$num is not a prime number"
fi
```

19. Find Duplicate Files

```
echo "Enter directory:"
read dirname
find "$dirname" -type f -exec ls -lh {} + | awk '{ print $5, $9 }' | sort |
uniq -d -w15
```

20. User Login Information

```
who
```