1. Print "Hello, World!"

A basic program to print text to the console.

```
echo "Hello, World!"
```

2. Display Current Date and Time

Use the `date` command to show system date and time.

```
date
```

3. List Files in a Directory

Use `ls` to list files in the current directory.

```
ls -l
```

4. Display Current Working Directory

Use `pwd` to show the current directory path.

```
pwd
```

5. Create and Remove a Directory

Use `mkdir` and `rmdir` to create and delete directories.

```
mkdir my_folder
rmdir my_folder
```

6. Create, Write, and Read a File

Use `echo` and `cat` to write and read a file.

```
echo "This is a sample text." > file.txt
cat file.txt
```

7. Use Variables in Shell

Assign and print variables.

```
name="Alice"
echo "Hello, $name!"
```

8. Take User Input

Use `read` to take input from the user.

```
echo "Enter your name: "
read name
echo "Welcome, $name!"
```

9. Simple If-Else Condition

Use an `if` condition to check a number.

```
echo "Enter a number:"
read num
if [ $num -gt 10 ]; then
  echo "Number is greater than 10"
else
  echo "Number is 10 or less"
fi
```

10. Loop Through Numbers (For Loop)

Use a `for` loop to print numbers.

```
for i in {1..5}
do
  echo "Number: $i"
done
```

11. Check if a File Exists

Use an `if` condition to check for a file.

```
echo "Enter filename:"

read filename

if [ -f "$filename" ]; then

  echo "File exists."

else

  echo "File does not exist."

fi
```

12. Append Text to a File

Use `>>` to append text without overwriting.

```
echo "This is additional text." >> file.txt

cat file.txt
```

13. Count the Number of Lines in a File

Use `wc -l` to count lines in a file.

```
wc -l file.txt
```

14. Check If a Directory Exists

Use an `if` condition to check for a directory.

```
echo "Enter directory name:"

read dirname
```

```
if [ -d "$dirname" ]; then
  echo "Directory exists."
else
  echo "Directory does not exist."
fi
```

15. While Loop Example

Print numbers from 1 to 5 using `while`.

```
i=1
while [ $i -le 5 ]
do
  echo "Count: $i"
  ((i++))
done
```

16. Case Statement Example

Use `case` to handle multiple choices.

```
echo "Enter a choice (start/stop/exit):"
read choice
case $choice in
  start) echo "System starting...";;
  stop) echo "System stopping...";;
  exit) echo "Exiting...";;
  *) echo "Invalid option";;
```

```
esac
```

17. Check if a Number is Even or Odd

Use modulus (%) to check even/odd.

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

18. Find the Factorial of a Number

Use recursion for factorial calculation.

```
factorial() {
  if [ $1 -le 1 ]; then
    echo 1
  else
    echo $(( $1 * $(factorial $(( $1 - 1 ))) ))
  fi
}
echo "Enter a number:"
read num
echo "Factorial of $num is $(factorial $num)"
```

19. Display Disk Usage

Use `df` to check disk space.

```
df -h
```

20. Print Environment Variables

Use `env` to list all environment variables.

```
env
```