Friday Challenge Tasks: Linux Scripts

Task 1

Create a new file called `task1.sh`.

Creata s simple script that will print out the following:

```
Hello, World!
```

Task 2

Create a new file called `task2.sh`.

Create a script that will print out the following:

```
Hello, World!
Today is: <current date>
You are logged in as: <username>
Your current working directory is: <current directory>
```

Task 3

Create a new file called `task3.sh`.

Create a script that will print out the following and uses the user's input to print out their name:

```
Hello, what is your name?
My name is: <name>
```

Task 4

Create a new file called `task4.sh`.

NOTE: You can either do it by asking the user for input or by hardcoding the numbers in the script.

Create a script that compares two numbers and prints out the following:

```
Enter the first number: NUMBER1
Enter the second number: NUMBER2
The first number is greater than the second number.
```

If the numbers are equal, print out the following:

```
Enter the first number: NUMBER1
Enter the second number: NUMBER2
The numbers are equal.
```

If the second number is greater than the first number, print out the following:

```
Enter the first number: NUMBER1
Enter the second number: NUMBER2
The second number is greater than the first number.
```

Task 5

Create a new file called `task5.sh`.

Create a script that will create as many files as the user specifies. The script should take the following arguments:

- The directory where the files will be created
- The number of files to create

Bonus: The files should be named `file1.txt`, `file2.txt`, `file3.txt`, etc. Don't hardcode the file names. Try to automate the process.

Task 6

Create a new file called `task6.sh`.

Create a script that asks you for an input after every file you created. When you type `yes`, the script creates the next file. When you type `no`, the script stops creating files.

Task 7

Create a new file called `task7.sh`.

Create a script that will ask the user for a file name and then print out the following:

```
What is the name of the file you are looking for?
The file <filename> exists.
```

If the file does not exist, print out the following:

```
What is the name of the file you are looking for?
The file <filename> does not exist.
```

Task 8

Create a new file called `task8.sh`.

Create a script that creates a backup of a directory. The script should take the following arguments:

- Source directory
- Destination directory

The script should copy all files from the source directory to the destination directory.

Bonus: Add a check to see if the destination directory exists. If it does not, create it. Name the backup directory with the current date.

Task 9

Create a new file called `task9.sh`.

Create a script that prompts the user for a directory path and then calculates and prints the total size of all files in that directory.

Task 10

Create a new file called `task10.sh`.

Create a script that asks you for a password and then checks if the password meets the following criteria:

- At least 8 characters long
- Contains at least one uppercase letter
- Contains at least one lowercase letter
- Contains at least one number
- Contains at least one special character

If the password meets all the criteria, print out the following:

```
Password is valid.
```

If the password does not meet all the criteria, print out the following:

```
Password is invalid.
```

BIG BONUS CHALLENGE

Task 11

Create a new file called `ec2-instance.sh`. Create a script that will create an EC2 instance in AWS. The script should take the following arguments:

- Instance type
- Key pair name
- Security group
- AMI ID
- Region

The script should use the AWS CLI to create the EC2 instance.

Task 12

Polish it even more and add the following:

- Add a check to see if the instance was created successfully.
- If the instance was created successfully, print out the following:

```
Instance created successfully.
```

- If the instance was not created successfully, print out the following:

```
Instance was not created successfully.
```

IDEA: It can be a good idea to create a script that will ask the user for the instance type, key pair name, security group, AMI ID, and region. Then it will create the EC2 instance using the AWS CLI.