Solution:

Exercise II:

```
#!/bin/bash


# Initialize variables
number=3
count=1
multiple=0


# Loop until the count reaches 11 (to get the first ten multiples)
until (( count > 10 ))
do
 # Calculate the multiple
 multiple=$((number * count))


 # Print the multiple
 echo "$number x $count = $multiple"


 # Increment the counter
 count=$((count + 1))
done


echo "First ten multiples of $number printed."
```

Exercise III:

```
#!/bin/bash

number=1

until (( number > 9 ))
do
  echo "$number"
  number=$((number + 2))
done

echo "Printed all odd numbers between 0 and 10."
```

Exercise IV:

```
#!/bin/bash

attempts=0
max_attempts=3
password=""
correct_password="Linux"

until [[ "$password" == "$correct_password" ]] || (( attempts >= max_attempts ))
do
  if (( attempts < max_attempts )); then
    read -p "Enter the password: " password
    if [[ "$password" == "$correct_password" ]]; then
```

```
    echo "Access granted!"
   else
    ((attempts++))
    echo "Incorrect password. Attempts remaining: $((max_attempts - attempts))"
   fi
  else
   echo "Too many attempts. Access denied!"
  fi
done
```

Exercise V:

```
#!/bin/bash

PS3="Please select an option: "

options=("Date" "Path" "Exit")

until [[ "$REPLY" -eq 3 ]]; do
  select opt in "${options[@]}"; do
    case "$opt" in
      "Date")
        date
        ;;
      "Path")
        read -p "Enter a directory path: " directory_path
        if [[ -d "$directory_path" ]]; then
          ls -l "$directory_path"
```

```
      else
        echo "Error: Directory '$directory_path' does not exist."
      fi
      ;;
    "Exit")
      echo "Exiting..."
      break
      ;;
    *)
      echo "Invalid option. Please try again."
      ;;
  esac
 done
done
```

Exercise VI.

```
#!/bin/bash

# Function to handle the interrupt signal (Ctrl+C)
interrupt_test() {
 echo -e "\nScript interrupted. Exiting now."
 exit 130 # Standard exit code for interrupted by Ctrl+C
}

# Trap the SIGINT signal and call the interrupt_test function
trap interrupt_test SIGINT
```

```
# Prompt the user to enter the filename
read -p "Enter the filename to check and potentially delete: " file_name

# Check if the filename is empty
if [ -z "$file_name" ]; then
  echo "Error: Filename cannot be empty."
  exit 1
fi

# Check if the file exists
if [ -e "$file_name" ]; then
  # Delete the file
  rm "$file_name"
  echo "File '$file_name' has been deleted."
else
  echo "File '$file_name' does not exist."
fi

# Sleep for 3 seconds
sleep 3

echo "Script finished."

exit 0
```