gcc program.c # Compiles program.c and generates a default executable named a.out **./a.out** # Runs the compiled program
gcc program.c -o my_program # Compiles program.c and creates an executable named my_program **./my_program** # Runs the compiled program
gcc program.c -o input_program # it takes inputs from input1.txt and displays the result in the terminal. **./input_program < input.txt**
gcc program.c -o io_program **./io_program < input.txt > output.txt** # it takes the inputs from input1.txt and creates the myoutput.txt document and writes the result here.
diff output1.txt myoutput.txt # compares the document output1.txt with the document myoutput.txt.
diff --ignore-all-space output1.txt myoutput.txt # compares the document output1.txt with the document myoutput.txt without ignoring the spaces.

QUESTION 1

A stationery store's product information is stored in a text file named urunler.txt, where each line represents one product. Each line contains the following information in order:

- Product code (integer),
- Product name (single-word string),
- Stock quantity (integer),
- Unit price (float number).

```
struct Product{

}
```

Write a C program that opens the **product.txt** file and reads the product information line by line. Identify the products with a stock quantity less than 10. Print these products to the screen and also write them to a new text file named **low_stock.txt**. Don't forget to perform error checking when opening and closing the files.

NOTE: The product information must be stored using a struct.

Input1: **product.txt file content**

```
1001 Pencil 5 12.5
1002 Notebook 20 25.0
1003 Eraser 8 3.5
1004 Ruler 15 6.0
```

Output1: **low_stock.txt file content**

```
1001 Pencil 5 12.50
1003 Eraser 8 3.50
```