PYTHON:-

sorted() function takes a list and returns the sorted list (obv) but we can customise how it sorts the list and this is where its important
list = sorted(list, key = lambda x: ____) in the _____ you write what the value upon which you are sorting is NOTE that if you want to sort by multiple criteria then make the lamda function return a tuple

FILES

reading files in python;-

```
with open("example.txt", "r") as file:
line = file.readline()  # Reads the first line as a string
while line: #keep executing until line is empty ie file is over
        print(line)
```

 OR use lines = file.readlines() => lines is a list of strings

OR content = file.read() => content is a string of the whole file content

How to read files with delimiter

```
with open(full_path,mode='r') as file:
        csvfile = csv.reader(file,delimiter='-')
        for line in csvfile:
```

Writing files in python:-

```
with open("example.txt", "w") as file:
   file.write("Hello, World!\n")
   file.write("This is a test file.\n")
```

```
lines = ["First line\n", "Second line\n", "Third line\n"]
with open("example.txt", "w") as file:
   file.writelines(lines)
```

Append to files:-

```
with open("example.txt", "a") as file:
   file.write("This line is appended.\n")
```

HOW TO READ COMMAND LINE ARGUMENTS

import sys first and then we can use sys.argv to get it

sys.argv is a list
sys.argv[0] is name pf python program
After that sys.argv[n] is nth argument

If the input is redirected from a file like for example

```
python3 playing_with_strings.py < q4-input.txt
```

THEN we must use sys.stdin

```
input_data = sys.stdin.read().splitlines()
```

then input_data[0] is first line in file and etc.
then we can use split() to get the individual arguments

NOTE god tier use of list comprehension ->

```
permuted_string = ''.join([string[i] for i in permutation])
```

OR
Took a dict and sorted based on the values

```
totalData = {k: list(v) for k, v in sorted(totalData.items(), key=lambda x: (-x[1][0], x[0]))}
```