Program description:

Program description is not described here due to confidentiality and must not uploaded on git repository. So, I assume the description of the problem is known.

Problem Statement:

Find the minimum number of MOs which are required to help the SO so that every machine at each site of the company is maintained. Create an application (CLI or HTTP API) that solves the problem.

Solutions: I have used two approaches. The logic remains the same for both approaches, hence only one solution (function) for both approaches. I have used two different ways to get the response from the REST API using GET and POST.

Solution 1 approach: GET Request (GetRequest Folder)

Step1: The program is written in Python3.8. Download the flask to run the program.

```
pip3 install flask
```

Step2: Go to the **folder- GetRequest. minimummachineoperators.py** contains the function for the above problem.

Step3: **flaskapiformachineoperators.py** contains the flask API program that fetch data from the above **minimummachineoperators.py**

Step4: make sure both the file is in the same folder and run the flaskapiformachineoperators.py program.

Step5: **Open a new terminal** and paste **curl** http://127.0.0.1:5000/input1 to check the output. One can change the end point like input2, input3, input4 and the values inside these end point can be changed.

Inputs:

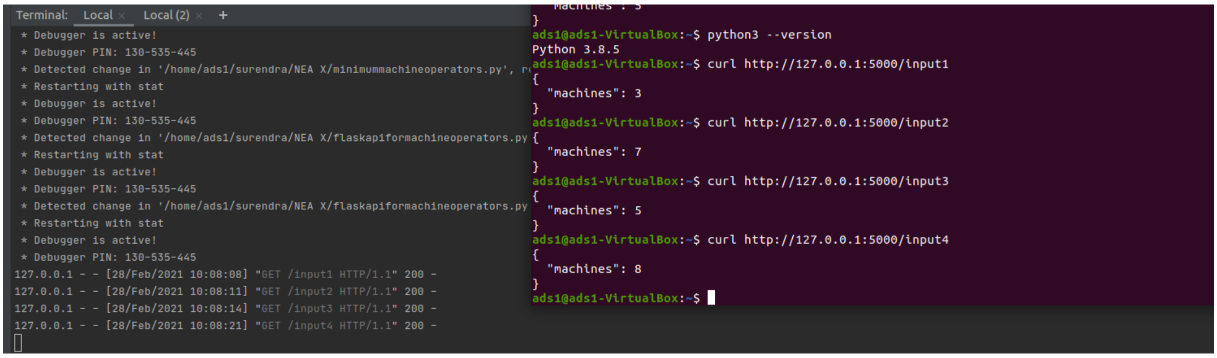
```
input1= {
    "machines": [15, 10],
    "C": 12,
    "P":5
}


input2= {
    "machines": [11,15,13],
    "C": 9,
    "P":5
}

input3= {
    "machines": [61, 10],
    "C": 50,
    "P":5
}

input4= {
    "machines": [40,10],
    "C": 10,
    "P":5
}
```

Output:



```
 * Debugger is active!
 * Debugger PIN: 130-535-445
 * Detected change in '/home/ads1/surendra/NEA X/minimummachineoperators.py', re
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 130-535-445
 * Detected change in '/home/ads1/surendra/NEA X/flaskapiformachineoperators.py
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 130-535-445
 * Detected change in '/home/ads1/surendra/NEA X/flaskapiformachineoperators.py
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 130-535-445
127.0.0.1 - - [28/Feb/2021 10:08:08] "GET /input1 HTTP/1.1" 200 -
127.0.0.1 - - [28/Feb/2021 10:08:11] "GET /input2 HTTP/1.1" 200 -
127.0.0.1 - - [28/Feb/2021 10:08:14] "GET /input3 HTTP/1.1" 200 -
127.0.0.1 - - [28/Feb/2021 10:08:21] "GET /input4 HTTP/1.1" 200 -
```

```
  "machines": 5
}
ads1@ads1-VirtualBox:~$ python3 --version
Python 3.8.5
ads1@ads1-VirtualBox:~$ curl http://127.0.0.1:5000/input1
{
  "machines": 3
}
ads1@ads1-VirtualBox:~$ curl http://127.0.0.1:5000/input2
{
  "machines": 7
}
ads1@ads1-VirtualBox:~$ curl http://127.0.0.1:5000/input3
{
  "machines": 5
}
ads1@ads1-VirtualBox:~$ curl http://127.0.0.1:5000/input4
{
  "machines": 8
}
ads1@ads1-VirtualBox:~$
```

==

Solution 2 approach: POST Request

Step1: The program is written in Python3.8. Download the flask to run the program.

pip3 install flask

Step2: Download the **postman** software for sending the post request in json format

Step2: Go to the folder- **PostRequest**. **minimummachineoperators.py** contains the function for above problem.

Step3: **flaskapiformachineoperators.py** contains the flask API program that fetch data from the above minimummachineoperators.py

Step4: Make sure both the file is in the same folder and **run the flaskapiformachineoperators.py** program.

Step5: **Open a postman** application, click on new API, change the request to POST and paste **http://127.0.0.1:5000/post/input**. Click on body, select the raw and give the text type as Jason and enter the input as below and click on send button.

```
{
    "machines": [11,15, 13],
    "C": 9,
    "P":5
}
```

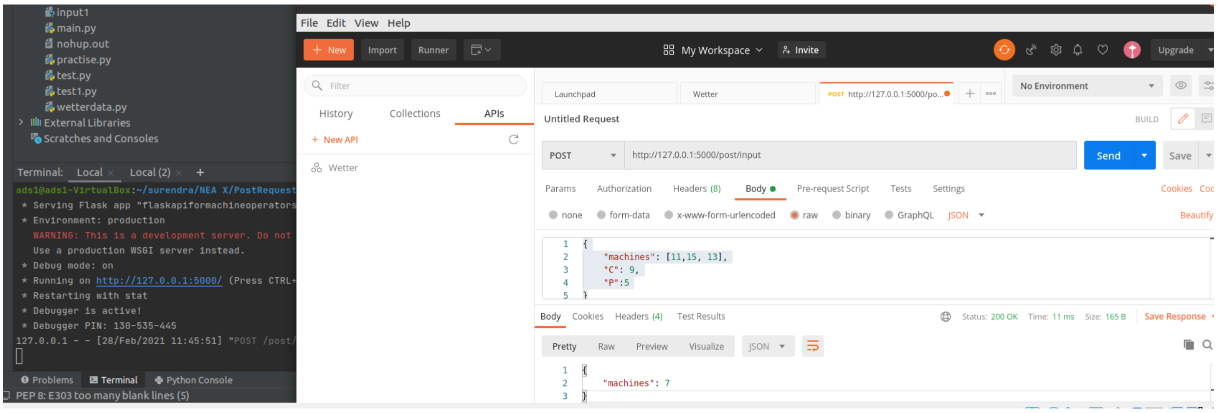
One can always change the value.

Step6: The output is displayed in the response section (bottom) of postman application.

```
{
    "machines": 7
}
```

Note: The second approach can be also used for GET request.

Input & Output:



```
ads1@ads1-VirtualBox:~/surendra/NEA X/PostRequest
 * Serving Flask app "flaskapiformachineoperators
 * Environment: production
   WARNING: This is a development server. Do not
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 130-535-445
127.0.0.1 - - [28/Feb/2021 11:45:51] "POST /post/
```

POST http://127.0.0.1:5000/post/input

Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "machines": [11,15, 13],
    "C": 9,
    "P":5
}
```

Status: 200 OK Time: 11 ms Size: 165 B

JSON

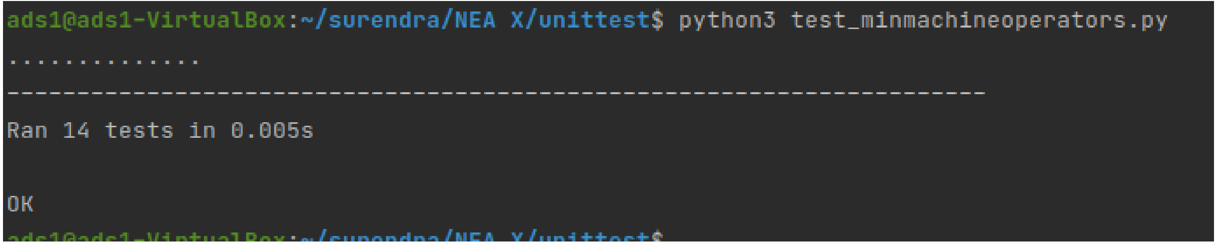
```
{
    "machines": 7
}
```

===

Unit test:

A unit test program is written to test different input values. Go to the folder- unittest. **test_minmachineoperators** is the unit test program and it also contains the function **minimummachineoperators.py.** I have used 14 tests to check the input conditions.

Output:



```
ads1@ads1-VirtualBox:~/surendra/NEA X/unittest$ python3 test_minmachineoperators.py
..............
----------------------------------------------------------------------
Ran 14 tests in 0.005s

OK
ads1@ads1-VirtualBox:~/surendra/NEA X/unittest$
```