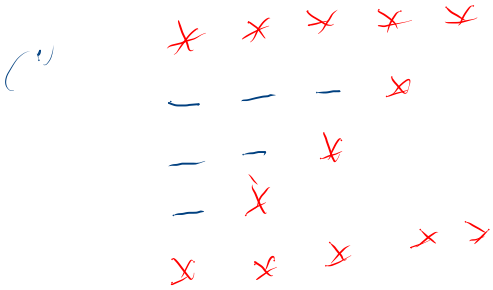
(ب)



(أ)

1. You are given as input marks of a student.
2. Display an appropriate message based on the following rules:
    2.1 for marks above 90, print excellent.
    2.2 for marks above 80 and less than equal to 90, print good.
    2.3 for marks above 70 and less than equal to 80, print fair.
    2.4 for marks above 60 and less than equal to 70, print meets expectations.
    2.5 for marks less than equal to 60, print below par.

Note -> Only change the code in area after - // code here

```
if marks > 90:
    print("excellent")
elif marks > 80:
    print
elif marks > 70
```

(91 – 100) x

(81 – 90) x

(71 – 80)

(DSP) → C, C++, Java → Python

int → 9 digit

long → 18 digit

10^{18}

input → string → (int) (1000 digit) (linked list)

→ input hour(24 hour format) and print good morning(<12) / good afternoon(<=16) / good evening(<=20) / good night.

① → input → num1
→ num2
→ operation

+ → num1 + num2
– → num1 – num2
* → num1 * num2

② → primenumber → 13 (1,13)
14 (1, 2, 7, 14)

Loop →

```
for i in range(7):
    print(i)
```
→ 0 1 2 3 4 5 6

```
for i in range(2,7):
    print(i)
```
→ 2 3 4 5 6

```
for i in range(2,7,2):
    print(i)
```
→ 2 4 6

$1 \le num \le 10^{12}$

$10^9 \rightarrow 1s$

$\frac{10^{12}}{2} \rightarrow \frac{1}{10^9} \times \frac{10^{12}}{2} = \frac{10^3}{2} s$

$= 8 min$ $1 \le num \le 10^{12}$

10^{100}

```
def isPrimeNumber(num):
    for i in range(2,num):
        if(num % i == 0):
            return False

    return True
```

10^9 operations → 1s (cpu)

10^{12} operations → $\frac{1}{10^9} \times 10^{12}$

$= 10^3 s \rightarrow$ 16 min

18/1 = 18
18/2 = 9
18/3 = 6
18/4 = 4
18/5 = 3
18/6 = 3

18/7 = 2
18/8 → 2
18/9 → 2
18/10 → 1.——
18/11 → 1.——
18/12 → 1

18/13 → 1.——
18/(20-18) → 1.——

n/x > 1

n/x ≥ 2

n/2 ≤ n

$num = p^a q^b$ (where p and q are prime numbers)

(1, 2, 3, 5, 7, 11 ----)

36 → 6·6 (6)
→ (2·3)(2·3)
→ (2·6)

17 → 1.17 (4)
15 → 3·5 (4)

num = a·b (a = b)

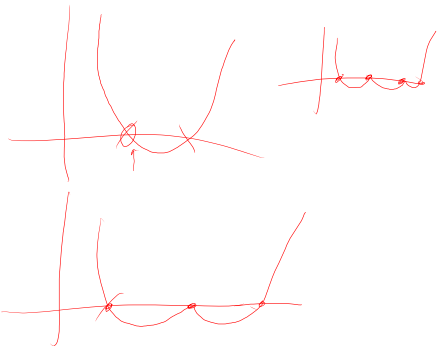
= a·a

$a \le \sqrt{num}$

$b \ge \sqrt{num}$

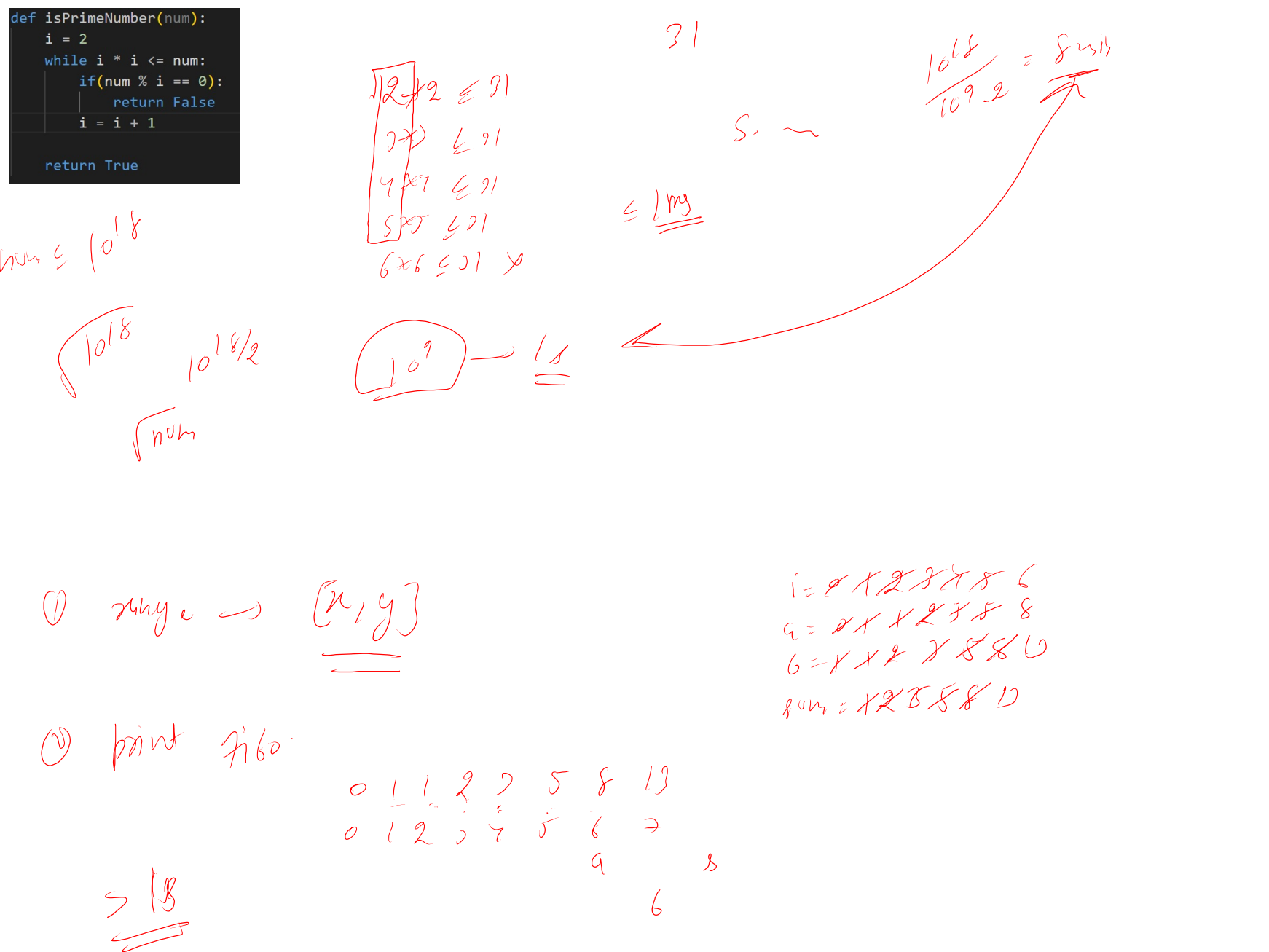
2 → 1¹ 2¹
3 → 1¹ 3¹
4 → 1² 2²
5 → 1¹ 5¹
6 → 2¹ 3¹
7 → 1¹ 7¹
8 → 1³ 2³
9 → 1² 3²
10 → 1¹ 2¹ 5¹
11 → 1¹ 11¹
12 → 1² 2² 3¹

13 → 1¹ 13¹
14 → 1¹ 2¹ 7¹
15 → 1¹ 3¹ 5¹



```python
def isPrimeNumber(num):
    i = 2
    while i * i <= num:
        if(num % i == 0):
            return False
        i = i + 1

    return True
```



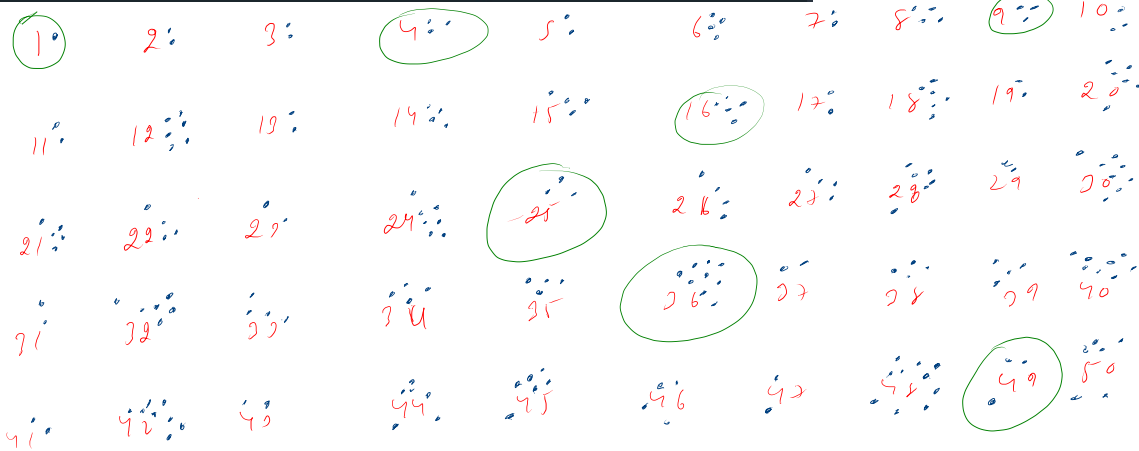
odd → ON
even → OFF

1. You are given n number of bulbs. They are all switched off. A weird fluctuation in voltage hits the circuit n times. In the 1st fluctuation all bulbs are toggled, in the 2nd fluctuation every 2nd bulb is toggled, in the 3rd fluctuation every 3rd bulb is toggled and so on. You've to find which bulbs will be switched on after n fluctuations.
2. Take as input a number n, representing the number of bulbs.
3. Print all the bulbs that will be on after the nth fluctuation in voltage.

n = 1000

(1, 1000)

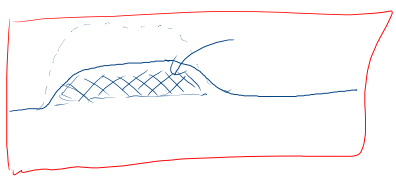
OFF
↓
ON
↓
OFF
↓
ON
↓
OFF



```
def bulbToggle(num):
    i = 1
    while i * i <= num:
        print(i)
```

Time:

CPU

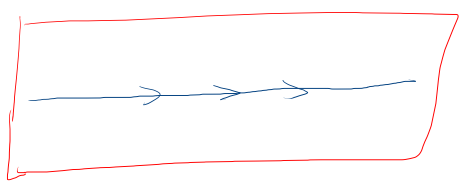


n = 1mil.

(Time + capacity)

Space:

RAM



1. You are required to print the Greatest Common Divisor (GCD) of two numbers.
2. You are also required to print the Lowest Common Multiple (LCM) of the same numbers.
3. Take input "num1" and "num2" as the two numbers.
4. Print their GCD and LCM.