1. How many zeros are there in 100!

   String str = “720”
   * 17

5 * 2 [100/5] + [100/25]

6! - X*Y = carry + result

2. Large File Sorting ->

   File with 10^10 numbers
   RAM can store only 10^5 numbers

4 6 8 2 5 7 | 4 6 8 2 5 7 10^10 - 10^5 -> 100000 chunks -> 50000

4 6 8 2

1123141421

1 2 3 4
6 7 8
2312431234234

10 ^ 5 + 10 ^ 5 TC : 0(10^5) * O(
While (i< A.size() && j< B.size())
A[i] < B[j]

-> which algorithm wil you use to sort this and how will you sort?

F1 f2 f3 f4 … f10^5

F1 f2 f3 - f3 f1 f2

Q. Client side rate limiter,

User -> your service -> 3rd party library(rate limited, 3rpm)

req	time
1	0.00s
2	0.05s
3	1.01s
4	1.03s (not allowed)
5	2.10s
6	2.80s
7	3.20s
8	3.90s
9	4.50s

0 - 2 sec - timestamp, value, 11:59-12:00 - 2 request per 1 minutes

key - {rate_limiting_11_58_12_00} - timestamp - 11:58 - 2

Key - {rate_limiting_locked} - 1 min after that will be TTL

Init - 2 redis calls, in between - 1, reached limit -

User -> client -> Rate limiter -> module

User, Server, RateLimitingManager, Library

1 2 3 req -

Counter, allowedRequests = 3;

If addedTimestamp == 0 {

	Counter = 0;

	addedTimestamp = currentTimestamp

```
        firstRecordTime = currentTimestamp
}
If (currentTimestamp - addedTimeStamp <= 1 minutes && counter <= allowedRequests
) {
        Counter ++;
        addedTimestamp = currentTimestamp
        flowAfterthat();
}
Else {
if(currentTimestamp - firstRecordTime > 1 minute)
        addedTimestamp = 0
        Counter = 0
}
}
```

Timestamp - 1

3 RPM

1st req - 0:00

2nd req - 0:05

3rd req - 0:58

4th req - 0:59

5th - 0:59

6th request - 1:00

7th req -