→ Pre-Computation :

Q) Given T test cases and in each test case a number N. Print its factorial for each test case % M,
Where $M = 10^9 + 7$

Constraints :
$1 \le T \le 10^5$
$1 \le N \le 10^5$

→
```
#include <bits/stdc++.h>
using namespace std;
const int M = 1e9+7;

int main()
{
    int T;
    cin >> T;

    while(T--)
    {
        int N;
        cin >> N;
        long long fact = 1;
        for(int i=2; i<=N; i++)
        {
            fact = (fact * i) % M;
        }
        cout << fact << endl;
    }
```

// Its time complexity is : $O(T * N)$ ⇒
If $T = N$ ∴ $O(N^2) \approx 10^{10}$
So, it will give TLE.

```
    RETURN 0;
}
```

// We can prevent our solution from TLE, by using precomputation.
Precomputation is storing values beforehand testing our Test cases.

→ Optimized Solution :

```
#INCLUDE <BITS/STDC++.H>
USING NAMESPACE STD;
CONST INT M = 1E9+7;
CONST INT N = 1E5+10;
LONG LONG FACT[N];

INT MAIN()
{
    FACT[0] = FACT[1] = 1;
    FOR (INT I = 2; I <= N; I++)
    {
        FACT[I] = (FACT[I-1] * I) % M;
    }

    INT T;
    CIN >> T;
    WHILE (T--)
```

```
{
    INT N;
    CIN >> N;
    COUT << FACT[n] << "\n";
}

// Its time complexity is O(N) + O(T) =
   10^5 + 10^5 = 2 * 10^5
It won't give TLE now.

RETURN 0;
}
```