Date ....

To calculate exact factorial We need to get exact number after each Multiplication.

Traditional Multiplication looks Like :-

example

```
   3 2 5 8 2 3 4 5  ] -> C
             x 2 3
 ------------------
   9 7 7 4 7 0 3 5  ] -> A
+6 5 1 6 4 6 9 0 0  ] -> B
 ------------------
 7 4 9 3 9 3 9 3 5  ] -> D
 ------------------
```

So What I did is Make a program that Multiply like this Using Arrays.

Let $n = 23$ (int type)

[It has 2 digits Therefore, we need two arrays to Store A & B (of int type)

[We need one More Array where C is Stored

Spiral

(int type) C = | 3 | 2 | 5 | 8 | 2 | 3 | 4 | 5 |

Now, $n = 23$

First take 3 :-

C ⇒

3	2	5	8	2	3	4	5

× 3

A ⇒

9	6	15	24	6	9	12	15

Now, each element of A should have one number

∴ Do this to each element :-

```
int x = A[i]
A[i+1] += x/10
```

(As it is int ∴ last Digit is omitted rest is added in A[i+1])

then A[i] = x % 10

(only last Digit is stored in A[i])

∴ A = | 9 | 7 | 7 | 4 | 7 | 0 | 3 | 5 |

Date .....

Now, for n=23

tak 2 :-

C ⇒ | 3 | 2 | 5 | 8 | 2 | 3 | 4 | 5 |

× 2

B ⇒ | 6 | 4 | 10 | 16 | 4 | 9 | 8 | 10 | 0 |

↑ Make it zero intentionally

Do same with every element of B :-

int x = B[i]

B[i+1] += x/10

B[i] = x % 10

∴ B = | 6 | 5 | 1 | 6 | 4 | 6 | 9 | 0 | 0 |

Date .....

Now, $D = A + B$

i.e. $\Rightarrow D[i] = A[i] + B[i]$

A ⇒

0	9	7	7	4	7	0	3	5

B ⇒

6	5	1	6	4	6	9	0	0

Add them :-

D ⇒

6	14	8	13	8	13	9	3	5

Do same with every element of D :-

int x = D[i]

D[i+1] = x/10

D[i] = x%10

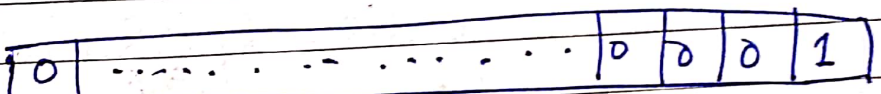
∴ D ⇒

7	4	9	3	9	3	9	3	5

Date .....

Now, Use this Multiplication Algorithm while Finding Factorial

initialize :-

C ⇒ 

Use Max Array size Alloted
(Cause in Factorial of Big numbers length of C can reach Big numbers like for 6000!
length should be 20066 atleast)

n = 2

We need Array C, D
& A, B, E, F, ...
(According to length of n)
for N=6000 (We need A, B, E, F i.e. 4 Arrays)

Date .....

Start loop :- or any no. of which factorial is Calculated

While ($n \leq 6000$)

{

Make every element of A, B, D = 0 to avoid garbage value
(Size of A, B, D, E, F should Also be Max. Array size Possible)

Use Multiplication Algorithm (explained previously) to C with n

C = D (i.e. Make every element of C equal to every element of D)

}

Display Array C