

Forum:[Hack Forums](#)

Thread: [C++ Data type bigger than long long?!](#)

Srgjan-LDTeam - August 16, 2014, 7:11 am

So im trying to solve a little problem, it looks like this:

Code "n! means $n \times (n - 1) \times \dots \times 3 \times 2 \times 1$

For example, $10! = 10 \times 9 \times \dots \times 3 \times 2 \times 1 = 3628800$,
and the sum of the digits in the number 10! is $3 + 6 + 2 + 8 + 8 + 0 + 0 = 27$.

Find the sum of the digits in the number 100!"

so when i call my function with parameter as value 10, my program returns 3628800,
so my code is right but there is another problem, when i call my function with parameter as value 100 which in this case (100 is needed), long long data type wont capture whole number in it, since the return-number is too big for long long,

What can i use to solve this problem? is there anything bigger than long long?

i already tried Code unsigned __int64 && unsigned long long

Sorry if its messed up a little :) and tnx in advance...

Mr. Dr. Prof. Patrick - August 16, 2014, 7:28 am

i do not think there is, the maximum an unsigned int in C/C++ is said to be able to hold is 4,294,967,295 or 0xffffffff and im sure the factorial of 100 is much bigger than that

but a part of being a programmer is solving your problems with what you have.. so i bet you can find a work around somehow.. so enjoy!

Masm4ever - August 16, 2014, 7:53 am

(Aug 16, 2014 07:28 AM)adv3nt Wrote: [\[CITE\]](#) i do not think there is, the maximum an unsigned int in C/C++ is said to be able to hold is 4,294,967,295 or 0xffffffff

While that is correct, it is irrelevant. A long long uses 64 bits, an unsigned int only 32.

dberube - August 16, 2014, 7:55 am

The only solution I see is using big integers. Their only limit in size is how much RAM you have.

A quick google search returned this: <https://mattmccutchen.net/bigint/>

Mr. Dr. Prof. Patrick - August 16, 2014, 8:01 am

(Aug 16, 2014 07:53 AM)Masm4ever Wrote: [\[CITE\]](#) While that is correct, it is irrelevant. A long long uses 64 bits, an unsigned int only 32.

yea i was making that comment with basic knowledge ive gotten poking around

Crypt0leucus - August 16, 2014, 10:06 am

(Aug 16, 2014 07:11 AM)Srgjan-LDTeam Wrote: [\[CITE\]](#) So im trying to solve a little problem, it looks like this:

Code "n! means $n \times (n - 1) \times \dots \times 3 \times 2 \times 1$

For example, $10! = 10 \times 9 \times \dots \times 3 \times 2 \times 1 = 3628800$,
and the sum of the digits in the number 10! is $3 + 6 + 2 + 8 + 8 + 0 + 0 = 27$.

Find the sum of the digits in the number 100!"

so when i call my function with parameter as value 10, my program returns 3628800,
so my code is right but there is another problem, when i call my function with parameter as value 100 which in this case (100 is needed), long long data type wont capture whole number in it, since the return-number is too big for long long,

What can i use to solve this problem? is there anything bigger than long long?

i already tried Code unsigned __int64 && unsigned long long

Sorry if its messed up a little :) and tnx in advance...

I have made an example for you using classes and a linked list representation of the big int. Maybe this will help you figure something out.

Download here: <https://www.dropbox.com/sh/a611q85kloblwvj/AAD4TWJNobnDaSjXXjzex8jDa>

6577657534735 - August 16, 2014, 10:57 am

Look up arbitrary precision arithmetic libraries

<https://gmplib.org/>

anikain - August 16, 2014, 7:24 pm

cough Float cough cough malloc cough

?lassy - August 17, 2014, 9:25 am

Last time I saw a guy solve this with C++ was with patterns.

What I mean by that is study the pattern of sums and the digits of the factorials from ex. 1 to 20.

Bruteforcing is not always the best way to go. I cannot say if that applies here or not, but keep it in mind when doing projecteuler.

0xDEAD10CC - August 17, 2014, 9:28 am

The other hack would be to store the known result in a string and add the numeric digits from characters extracted from the string, but technically this would still be cheating since your program didn't calculate the result of that factorial. :)