REGULATIONS

Due date: 06 April, 23:59, Monday (Not subject to postpone)

Submission: Electronically. You will be submitting your program source code written in a file which you will name as the1.c through the COW web system. Resubmission is allowed (till the last moment of the due date), The last will replace the previous.

Team: There is no teaming up. The take home exam has to be done/turned in individually.

Cheating: This is an exam: all parts involved (source(s) and receiver(s)) get zero+both parts will be subject to disciplinary action.

PROBLEM

One way to represent integral numbers is using a fixed base system. Here, as we all know, numbers are represented by digits where the $n^{th}$ position (here position count $n$ starts at the right most with a 0 value) has a place value of $base^n$. Our decimal system is such a system where the base is 10.

Another, but less preferred way to represent integers is to use place values that obey different rules. One can think of several alternatives one of which is using factorial as the place value. In other words the $n^{th}$ position (position counting start from right most with 1 as a value) has a place value of $n!$. Of course any position with a place value of $n!$ will contain at most a digit of value $n$. We call this system Factorial Base System and abbreviate it as FBS.

Here is an example:
The decimal value 6592 is expressed in this system as 1204220 meaning:

$$6592 = 1 \times 7! + 2 \times 6! + 0 \times 5! + 4 \times 4! + 2 \times 3! + 2 \times 2! + 0 \times 1!$$

A second example reads as:
The decimal value 23458783475780 is expressed in this system as B35C739472452200. Here B and C are digits representing the values 11 and 12 respectively.

SPECIFICATIONS

- You are expected to write a program that reads from standard input and outputs to standard output.

- The input is a single decimal integer $> 0$ given in a single input line. The line termination can be by an EOL character or directly by an EOF (Both cases are possible!). The maximal size of this integer is so that it can be expressed as described in the following item.
• The output is a number expressed in FBS with at most 35 positions. For a number having all the 35 positions full, the leftmost digit has the place value of 35! and the rightmost digit has the place value of 1!.
  
  As it is with decimal numbers trailing zeros will not be printed.
  
  No white space characters will exist between the positions.

• Digits that have values $\geq 10$ will be represented by single letters from the English alphabet. So, the values 10, 11, $\ldots$, 35 will be represented by A, B, $\ldots$, Z, respectively.

• There will be no erroneous input. Don’t waste your time on error checking.

• No dynamic memory allocation. No string manipulation (you do not need it).

• If you decide that you have to use arbitrary size integer arithmetic you cannot use code from the internet. Doing so will be considered cheating. If you need it you are supposed to code it by yourself.

• You are not allowed to make OS shell calls (if you do not know what this means leave it so).

• Do not beautify your input and/or output.

EVALUATION

• A program that produces the correct result for a single test case will be considered a ‘working program’ and will receive 50 points. If a program produces correct results for all test cases it will receive 100 points.

• A non working program will be graded by the teaching assistant visually (this grading is decisive and not open to objection nor explanation).

• All submissions will be compiled and run multiple times under strictly equal conditions on multiple data inputs.