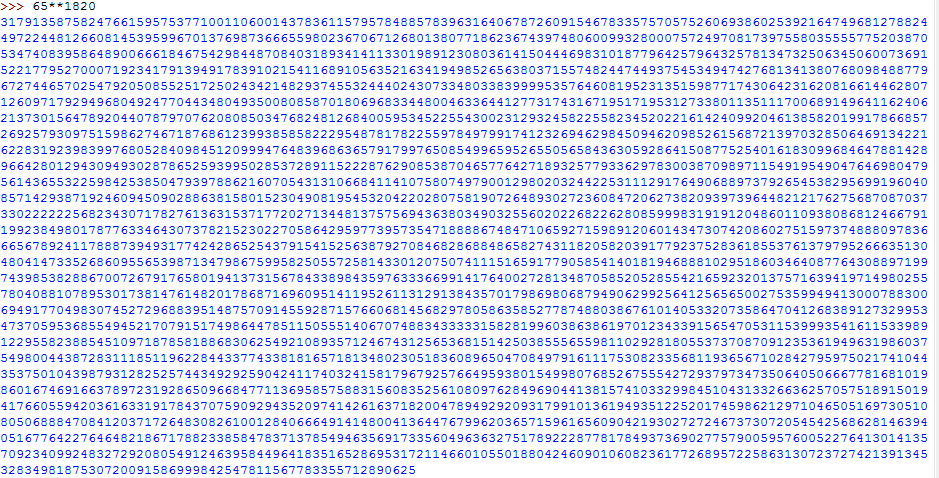
1. **Using the exponential notation \*\*, express the number of different possible strings of these characters that would constitute a line. Assume that a line has 70 characters (including spaces) and that there are 65 possibilities for each character. Calculate the number using the python Idle shell.**
   1. There are 65 possible characters possible in a space in the line, and 70 character per line, so there are 65\*\*75 possible permutations for characters, which is: 8015580221908051830252864620266291701029887523957704759968266340081437847271082289810940901819691362106823362410068511962890625. Which is a very large number of possible lines!
   2. Source:
2. **Using the exponential notation \*\*, express the number of different possible strings of characters that would constitute a page. Assume that there are 26 lines per page (70 chars/line × 26 lines/page = 1820 characters). Calculate the number using the python Idle shell.** 
   1. Since there are 1820 characters on a page, and 65 possible choices for characters this is represented as 65\*\*1820. To check also note that 65\*\*75 which is the number of random character permutations per line, raised to the number of lines gives the number of random lines per page e.g. 65\*\*75\*\*26, which is equivalent to 65\*\*(75\*26) = 65\*\*1820, so our answer makes sense. 65\*\*1820 is: a very large number. Below is an image of the calculation:
3. **Using the exponential notation \*\*, express the number of different possible strings of characters that would constitute a book of this size. Try to calculate this number using the python interpreter. What happens? (I know that the answer has 620,307 digits. [Thanks for Jordan MacQueen for the correction.] Bonus point: How do I know that?) **
   1. Since there are 188 pages, again we must raise 65\*\*1820\*\*188 = 65\*\*(1820\*188) which is an incredibly large number.
   2. This causes my idle shell to crash.
   3. We can determine the number of digits of 65\*\*(1820\*188) by clever use of the Logarithm function. If we take the log of this number using a base of 10 we can calculate how many multiples of 10 go into it. I.e., the digits in our base 10 number system. This requires the python math module. Below is the code proving the result, the extra .4 or so, is in the 1’s place, so it counts a digit, giving 620,307:

