**Factorial**

Problem Statement:

Find the factorial of any integer number. The numbers could be at most ten-digit long.

* Input example:

Enter number: 1000

* Output Example:

Factorial is: 402387260077093773543702433923003985719374864210714632543799910429938512398629020592044208486969404800479988610197196058631666872994808558901323829669944590997424504087073759918823627727188732519779505950995276120874975462497043601418278094646496291056393887437886487337119181045825783647849977012476632889835955735432513185323958463075557409114262417474349347553428646576611667797396668820291207379143853719588249808126867838374559731746136085379534524221586593201928090878297308431392844403281231558611036976801357304216168747609675871348312025478589320767169132448426236131412508780208000261683151027341827977704784635868170164365024153691398281264810213092761244896359928705114964975419909342221566832572080821333186116811553615836546984046708975602900950537616475847728421889679646244945160765353408198901385442487984959953319101723355556602139450399736280750137837615307127761926849034352625200015888535147331611702103968175921510907788019393178114194545257223865541461062892187960223838971476088506276862967146674697562911234082439208160153780889893964518263243671616762179168909779911903754031274622289988005195444414282012187361745992642956581746628302955570299024324153181617210465832036786906117260158783520751516284225540265170483304226143974286933061690897968482590125458327168226458066526769958652682272807075781391858178889652208164348344825993266043367660176999612831860788386150279465955131156552036093988180612138558600301435694527224206344631797460594682573103790084024432438465657245014402821885252470935190620929023136493273497565513958720559654228749774011413346962715422845862377387538230483865688976461927383814900140767310446640259899490222221765904339901886018566526485061799702356193897017860040811889729918311021171229845901641921068884387121855646124960798722908519296819372388642614839657382291123125024186649353143970137428531926649875337218940694281434118520158014123344828015051399694290153483077644569099073152433278288269864602789864321139083506217095002597389863554277196742822248757586765752344220207573630569498825087968928162753848863396909959826280956121450994871701244516461260379029309120889086942028510640182154399457156805941872748998094254742173582401063677404595741785160829230135358081840096996372524230560855903700624271243416909004153690105933983835777939410970027753472000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000

Algorithm:

IMPORT sys

SET The recurtion limit to 10000

DEFINE FUNCTION fac(n):

if(n<2):

RETURN "1"

RETURN multiply(str(n),fac(n-1))

DEFINE FUNCTION multiply(num1, num2):

SET res TO 0

SET carry1 TO 1

FOR n1 IN num1(REVERSE ORDER):

SET carry2 TO 1

FOR n2 IN num2(REVERSE ORDER):

res += int(n1)\*int(n2)\*carry1\*carry2

carry2 \*= 10

carry1 \*= 10

RETURN str(res)

IF \_\_name\_\_ EQUALS "\_\_main\_\_":

num=int(INPUT("Enter number: ",))

OUTPUT("Factorial is: ",CALL fac(num))

Proposed Python Code:

/\* ------- main.py ------- \*/

import sys

sys.setrecursionlimit(10000)

def fac(n):

if (n < 2):

return "1"

return multiply(str(n), fac(n-1))

def multiply(num1, num2):

res = 0

carry1 = 1

for n1 in num1[::-1]:

carry2 = 1

for n2 in num2[::-1]:

res += int(n1)\*int(n2)\*carry1\*carry2

carry2 \*= 10

carry1 \*= 10

return str(res)

if \_\_name\_\_ == "\_\_main\_\_":

num = int(input("Enter number: ",))

print("Factorial is: ", fac(num))

/\* ---------------------- \*/

Conclusion:

The proposed algorithm has a runtime of O(n2), where n is the length of the number.

Limitations and assumptions for this algorithm include:

1.The program will only create permutations of three digits numbers and some four digits number.