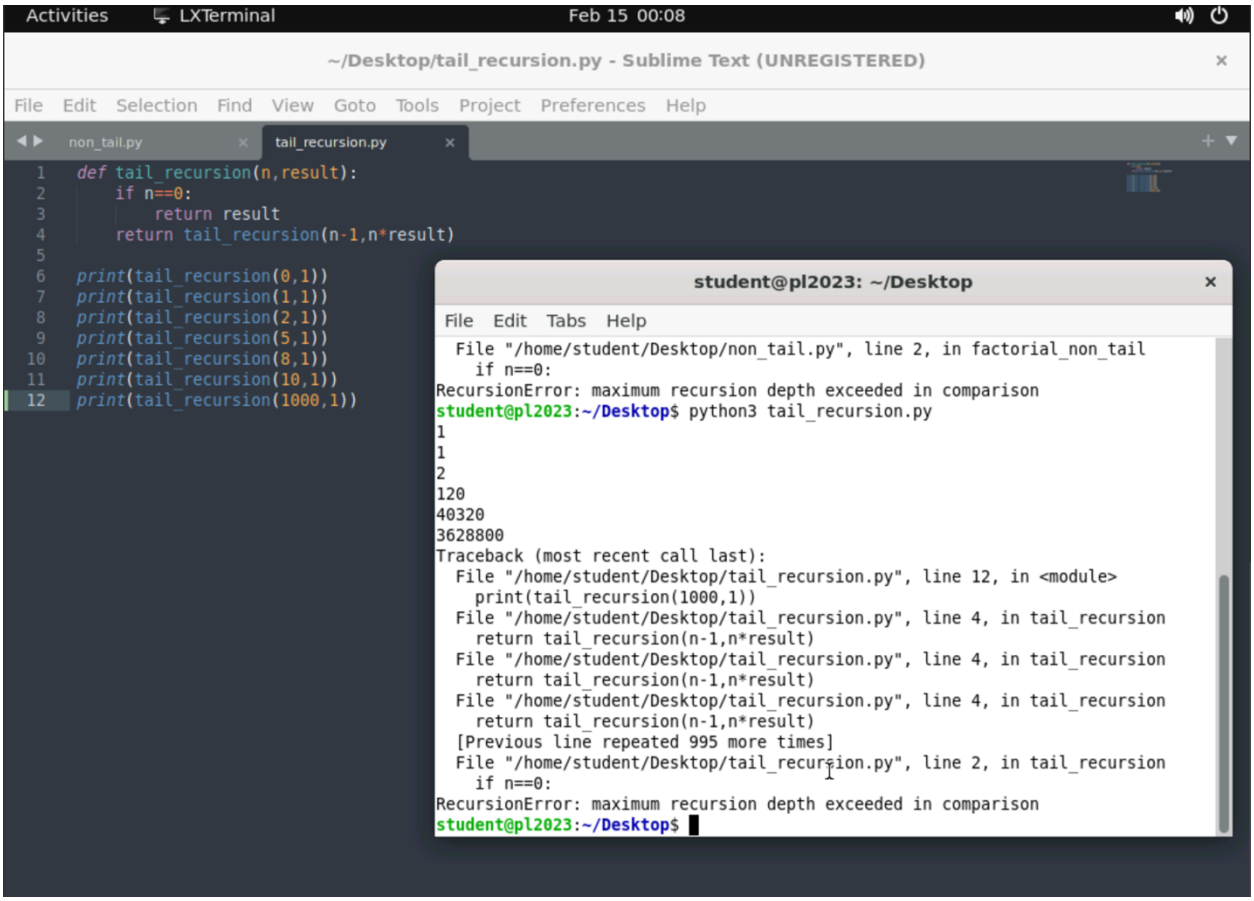
Tail Recursion

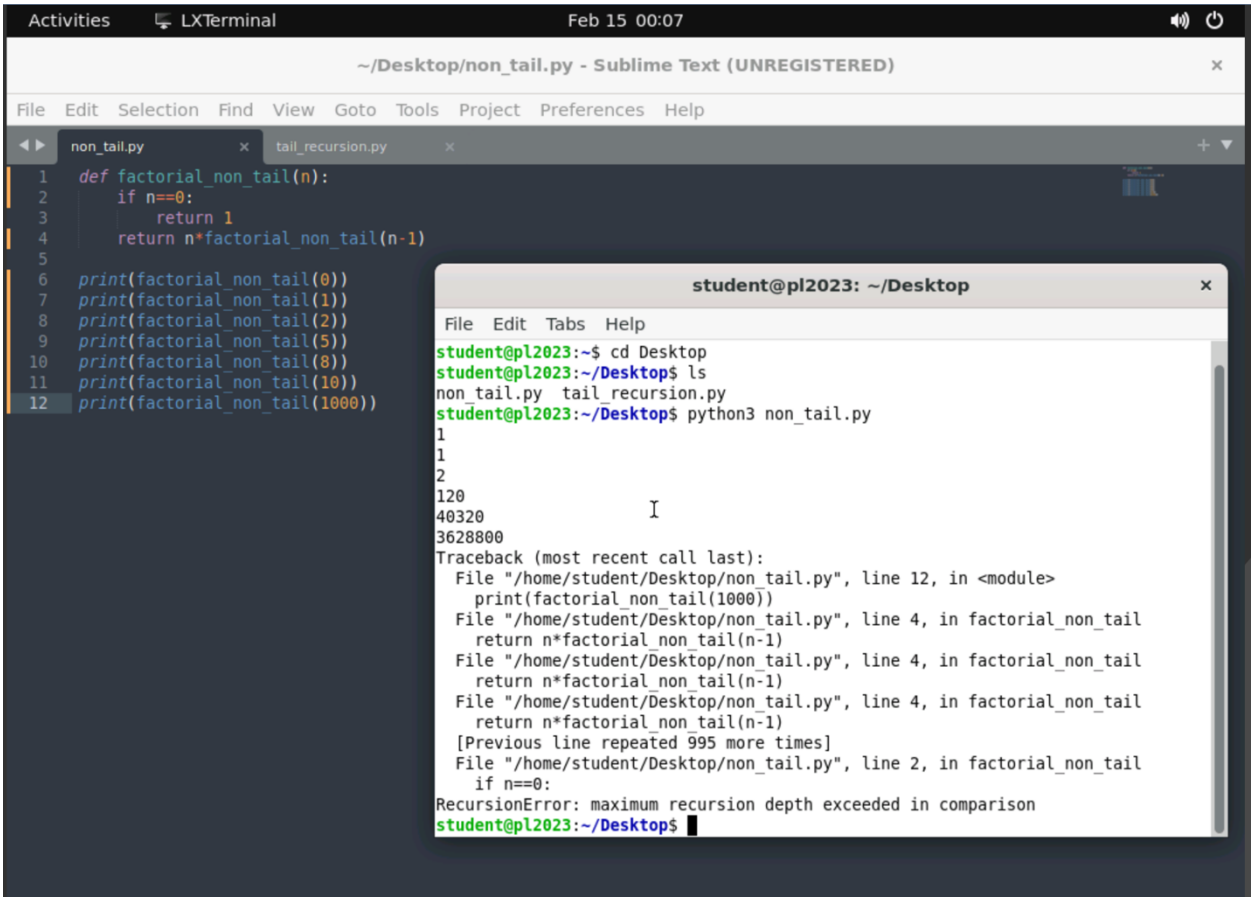
In SBCL tail-recursive and non-tail-recursive version were able to handle larger inputs as compared to Python because in python we eventually hit system limits on the stack size which depicts that SBCL supports tail recursive optimization whereas Python doesn't.
However, in SBCL when a large number is given as input the non-tail version leads to stack-overflow error first because each recursive call consumes additional stack space to maintain context until its recursive call is resolved. In contrast, a tail recursive function minimizes stack usage by reusing the stack frame for successive calls.

Python

Tail recursion

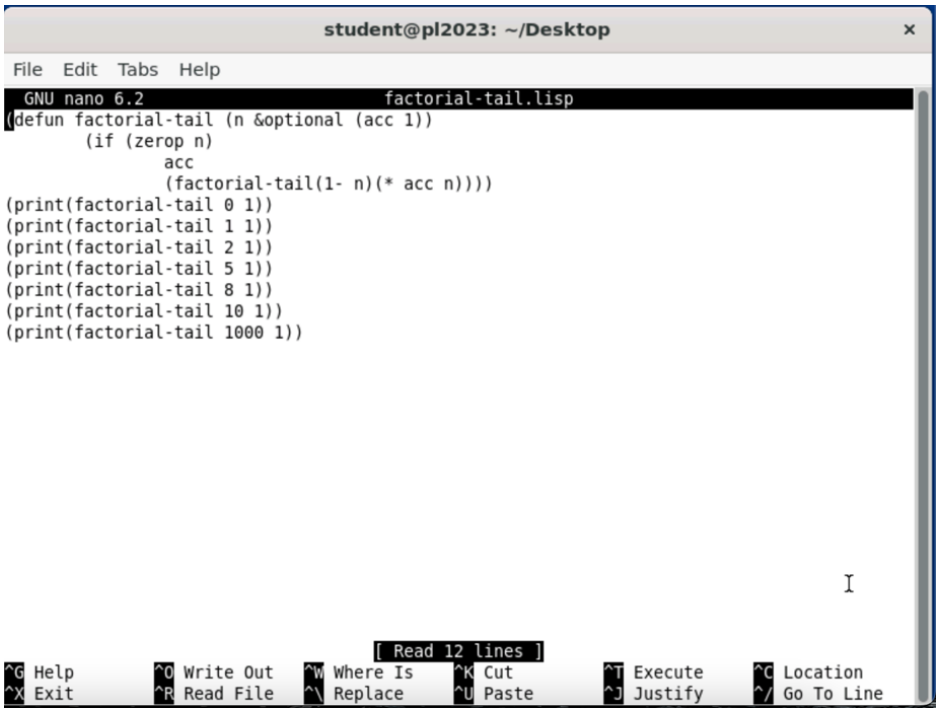


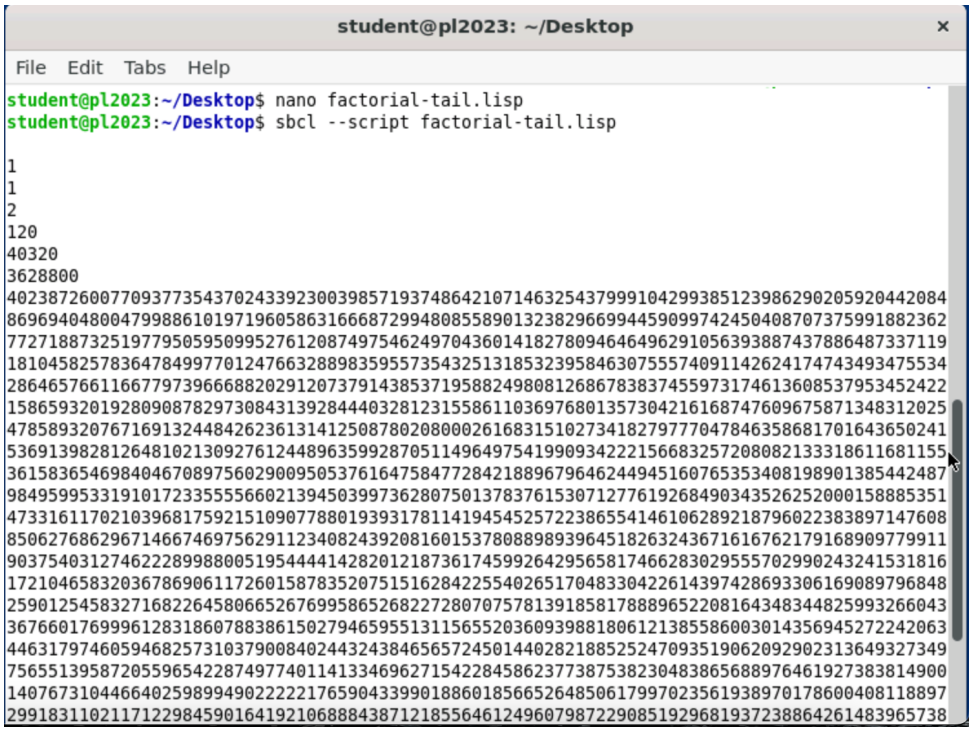
Non-tail recursion



Lisp

Tail recursion

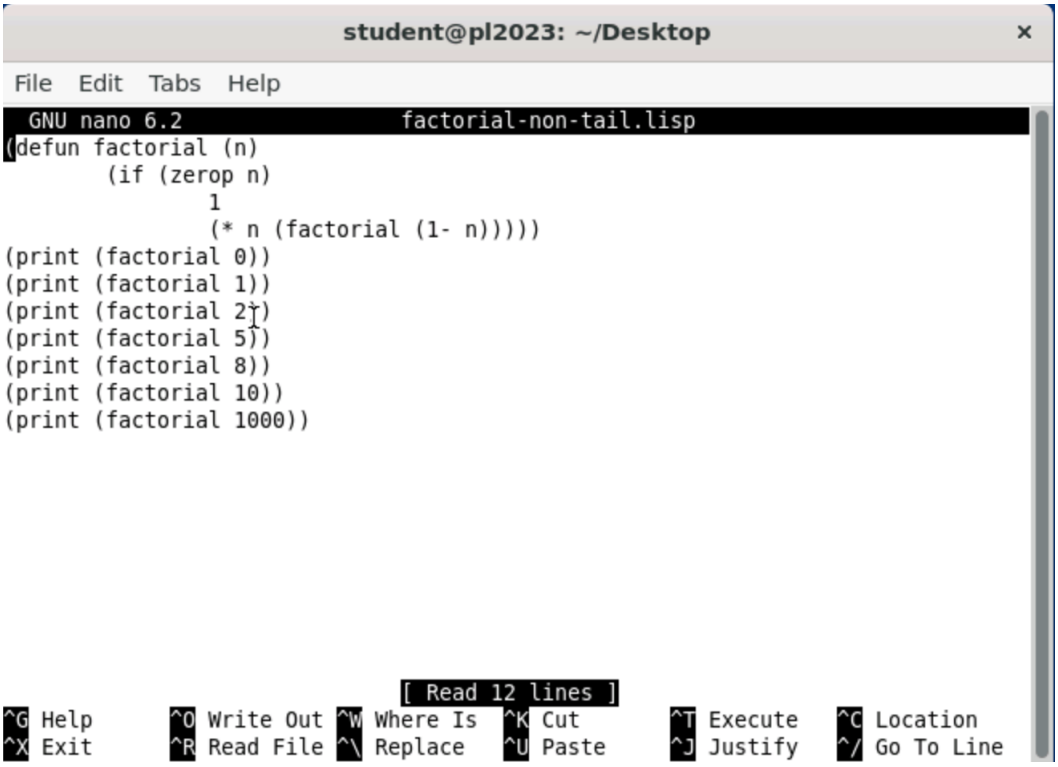


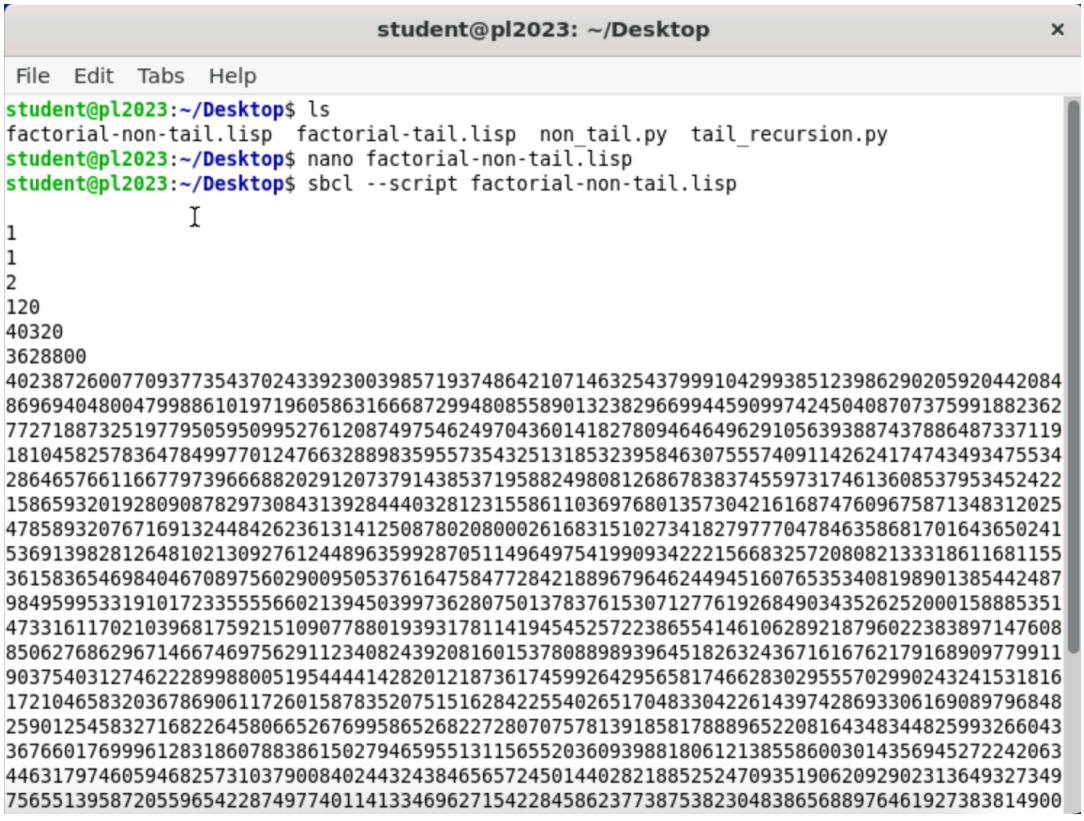



student@pl2023: ~/Desktop
File Edit Tabs Help
GNU nano 6.2 factorial-tail.lisp
(defun factorial-tail (n &optional (acc 1))
(if (zerop n)
acc
(factorial-tail(1- n)(* acc n))))
(print (factorial-tail 100000))
[Read 6 lines]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo

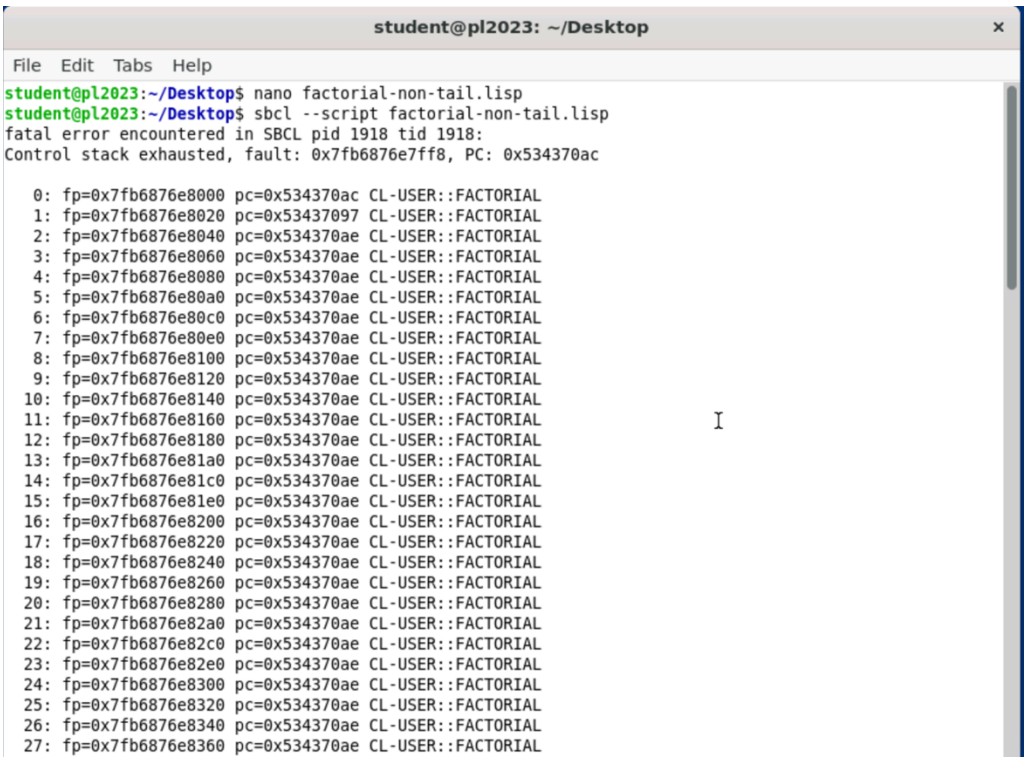

student@pl2023: ~/Desktop
File Edit Tabs Help
4780143381420990984675760752150116926505118145642185502501818401376907625095660181499319940308652888426182272323130451995919490543044644920205573769871961590155894027993494118223028371875967119907783155279025695259092576950649117833636803393005023815868315789579340175537825348671345300128777866703065105359337466212676185491451331031629652314978515685792850759313410480960600044510380261154696157823302425336728711773607212355035708819795804341280203100792174021456249062309906350680593395981377034317675382664169044153245508707856539712789524976146630856572706468609349277847437120670927760157496568414217978277963253691288825894095566543884980682688245660248731825916368580749796308448355436406190817816384463686490692185135204262669397356766941876264937942091286821962530753613845534896750884074062053970805637093433337297055959199347959700533772012423147410604739905552209757625227782043041392707851759199091165254149547648088670732282187581146148281480074568616459130919280706019860515139535141429840967898431147883259732646203098818559811008099548739490541826287806276069779990247360569313736105442595209970008483126219592455102578135012900220576967612689364147210519692052253710118420060533921087863513548531861186122383542242845070413947643319041060070981786775275806325890311868054040675779524843733298028515528433650071717042680086516275518241043390100842336559049726669451765822910434497748416539367770051599757734187546648801632479825742217034235938716266826261974143706081257129373804563951703004584503961354775323774100820601744310556920775304370337102163102738621831178697477743952019448216335544020464076369934855797248383157119927231291448723059456232658510548396630867224254027308580508719620185959953380168700632900109801774120748515586342773813048907876202002421342447902574624378556771443574113845297932489341822877696896287902719035128564914982223738493903965622446279976545922043319273674631931670364909881844153056396448204922534599108844436132556521778073860489039917326320142262595027565899327717924599108416859586700982058386548872743719388391741207461744996230795007749207892163259081015862461449201422124524364330022162359682563604743157957064406520908530522816379893246087703188966936785242355961298634047783902164102641812666343622647457318293894090477519310441299124952077392690851662476821095904911719124549684524204074555566761330545858680633005626362771760748318036130492740373573972587432747615048944183554513491397000579073436073353245120204476005232191628156267538510691898385565710531023575053977933160146925558121267658750984206642170595573078442253473572251073584002338891695032608869018118450092434353813660124636530621825414964713352184522292531214059474299371885318327979643726350903979292986917730236762400453803952052911013000267850264565528552559530143481153537011520228661876562016645889101326934613616325636728083050851299046922288515208325877297581424974819874963262749205633309127026011797096728439843127934664596136238900227177517713218215086433114260479230653785776203117168503677682167908513333392196823009837942574930747620308738432826152874172863277012812337571490908498310634068776935960696170157614649581402824388050132300840236556412359722149531096990212346170384338130057578946736621734918137930716509698157327325273831689460048459609422810580147808902992842927437011051332236420266951613151417189610359134933989995224688154409516413711099811575287891190347829272904357935

Non-tail recursion






student@pl2023: ~/Desktop
File Edit Tabs Help
GNU nano 6.2 factorial-non-tail.lisp
(defun factorial (n)
(if (zerop n)
1
(* n (factorial (1- n)))))
(print (factorial 100000))
[Read 5 lines]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo


student@pl2023: ~/Desktop
File Edit Tabs Help
student@pl2023:~/Desktop$ nano factorial-non-tail.lisp
student@pl2023:~/Desktop$ sbcl --script factorial-non-tail.lisp
fatal error encountered in SBCL pid 1918 tid 1918:
Control stack exhausted, fault: 0x7fb6876e7ff8, PC: 0x534370ac
0: fp=0x7fb6876e8000 pc=0x534370ac CL-USER::FACTORIAL
1: fp=0x7fb6876e8020 pc=0x53437097 CL-USER::FACTORIAL
2: fp=0x7fb6876e8040 pc=0x534370ae CL-USER::FACTORIAL
3: fp=0x7fb6876e8060 pc=0x534370ae CL-USER::FACTORIAL
4: fp=0x7fb6876e8080 pc=0x534370ae CL-USER::FACTORIAL
5: fp=0x7fb6876e80a0 pc=0x534370ae CL-USER::FACTORIAL
6: fp=0x7fb6876e80c0 pc=0x534370ae CL-USER::FACTORIAL
7: fp=0x7fb6876e80e0 pc=0x534370ae CL-USER::FACTORIAL
8: fp=0x7fb6876e8100 pc=0x534370ae CL-USER::FACTORIAL
9: fp=0x7fb6876e8120 pc=0x534370ae CL-USER::FACTORIAL
10: fp=0x7fb6876e8140 pc=0x534370ae CL-USER::FACTORIAL
11: fp=0x7fb6876e8160 pc=0x534370ae CL-USER::FACTORIAL
12: fp=0x7fb6876e8180 pc=0x534370ae CL-USER::FACTORIAL
13: fp=0x7fb6876e81a0 pc=0x534370ae CL-USER::FACTORIAL
14: fp=0x7fb6876e81c0 pc=0x534370ae CL-USER::FACTORIAL
15: fp=0x7fb6876e81e0 pc=0x534370ae CL-USER::FACTORIAL
16: fp=0x7fb6876e8200 pc=0x534370ae CL-USER::FACTORIAL
17: fp=0x7fb6876e8220 pc=0x534370ae CL-USER::FACTORIAL
18: fp=0x7fb6876e8240 pc=0x534370ae CL-USER::FACTORIAL
19: fp=0x7fb6876e8260 pc=0x534370ae CL-USER::FACTORIAL
20: fp=0x7fb6876e8280 pc=0x534370ae CL-USER::FACTORIAL
21: fp=0x7fb6876e82a0 pc=0x534370ae CL-USER::FACTORIAL
22: fp=0x7fb6876e82c0 pc=0x534370ae CL-USER::FACTORIAL
23: fp=0x7fb6876e82e0 pc=0x534370ae CL-USER::FACTORIAL
24: fp=0x7fb6876e8300 pc=0x534370ae CL-USER::FACTORIAL
25: fp=0x7fb6876e8320 pc=0x534370ae CL-USER::FACTORIAL
26: fp=0x7fb6876e8340 pc=0x534370ae CL-USER::FACTORIAL
27: fp=0x7fb6876e8360 pc=0x534370ae CL-USER::FACTORIAL