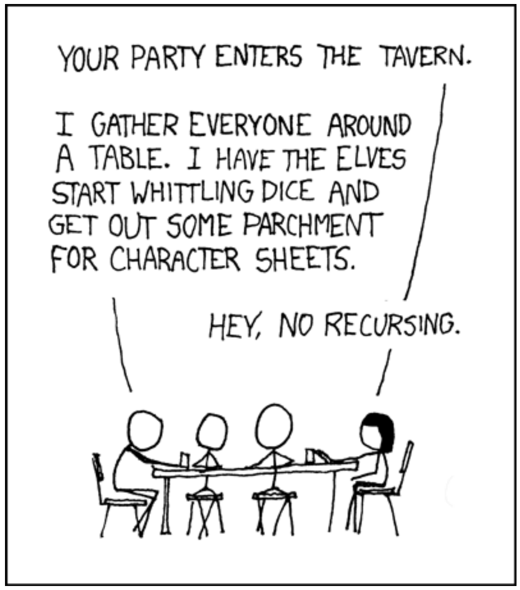
CS52 - Assignment 4

Due Monday 2/29 at 11:59pm



https://xkcd.com/244/

Getting Started

- For this assignment we will be coding up functions in CS41B. Make sure you've read the CS41B manual on the course web page and refer to it as you work through this assignment.
- Each problem will be in it's own file, so you will be creating four files: `positive.a41`, `power2.a41`, `ackermann.a41` and `smash.c`.
- Refer to section A of the CS41B handout for information about running the CS41B machine. If you're working on your own laptop, you can obtain a copy from:

  `http://www.cs.pomona.edu/~dkauchak/classes/s16/cs52-s16/handouts/cs41b.jar`

  Save this in the same folder as your `.a41` files to make your life simpler.
- *Aquamacs note*: Often Aquamacs uses a variable-width font which makes it difficult to line up columns. To force it to use a fixed-width font, go to

  `Options` ⟶ `Appearance` ⟶ `Auto Faces`

  and uncheck `Auto Faces`.

Optional: Touching base

We're about a third of the way through the course and I wanted to touch base with you all, see how the course is going and see if there is any way we can make it better. I have created a *very* short survey. Please take 5 minutes and fill out the anonymous survey by the assignment due date (optional). I will briefly go over the results in class.

`https://docs.google.com/forms/d/1_CXui0Zx_nxu50D9r8v8x42NQlcGPutCEGMG1iIa0Ak/`

A note on debugging in CS41B

Eventually (hopefully already!) you'll get the hang of the CS41b language, the stack and how to make function calls. Once you reach this stage, if all goes well, you'll code up your functions and with minimal tweaking, they'll work!

While I try and remain optimistic, most of you will still probably run into one last bug or two (I almost always do!) that you can't figure out by just looking at the code. If you reach this point and you really understand what's going on in your code, but can't find the bug, then the next step is to step through the execution of your code and follow the execution. If you do this a step at a time, you'll eventually find the problem.

A few thoughts on this:

– Open your code in the simulator and run it a line at a time using the "step" button.

– Get out a piece of paper and write down a place for r2, r3 and the stack. As you execute each line of code, update what you think should be happening on the piece of paper and then double check that's what happened in the simulator. If they're different, figure out why!

– Some of the instructions that you write show up, and are executed as, two instructions. In particular,

  - `psh`: is a store based on r1 (sto) followed by a decrement of r1 (sbc)
  - `pop`: is a load based on r1 (loa) followed by an increment of r1 (adc)
  - `lcw`: loads the lower bits (lcl) and loads the higher bits (lch)
  - `jmp`: is a load (loa) followed by a function call (cal)

  Besides these four, there will be a direct one-to-one mapping between the instructions you write and the instruction view on the right. Don't get thrown off by this. You can still easily step through an instruction at a time and see your code execute.

– Remember, r1 holds the location of the next location where a value would be put on stack. To view the stack:

  1) Look at the memory location stored in r1, i.e. the value in r1.

2) Look at the data view on the left of the simulator and find the address (left part before the colon) corresponding to the address in r1.
3) The address in r1 is not part of the stack. The first value in the stack is the address immediately below the address in r1 (e.g. if r1 is 00da then the first value of the stack is 00dc). Remember, the stack will grow towards smaller memory addresses, so the stack itself will be in the larger memory addresses.

Using these things, you should be able to step through your code a line at a time. If you understand what's supposed to be happening, then this can be very, very helpful at identifying small bugs.

When I first wrote Ackerman, I had a small bug in my unction (I was doing a "`loa r2 r1 6`" when it should have been"`loa r2 r1 4`") and I only found it by doing what I describe above. I know it may seem a bit tedious, but I promise you will find the bug much, much faster using this approach than most others.

1 Warming up

Write a CS41B program `positive.a41` that:

- Prompts the user for a number
- then prints 1 if the number is positive, 0 otherwise.
- You *must* include in your program a separate function that takes a single parameters as "input" and "returns" 1 if the number is positive, 0 otherwise.

For example, if we ran the program in the simulator and input 10 we would see the following:

```
CS41B wants a value > 10
CS41B says > 1
```

2 If you push, you must pop

Write a CS41B program `power2.a41` that takes as input a number n and prints out 2^n. To accomplish this write a function that computes powers of 2 by following the recursive pattern suggested by the SML function below.

```
fun power2 k =
     if k < 0 then
          0
     else if k = 0 then
          1
     else
          double(power2(k-1));
```

"Doubling" may be implemented by adding a number to itself; there need not be a separate doubling function. On the other hand, the `power2` function itself *must* be implemented recursively. That is, the function must save the return address on the stack and eventually jump back to it, and the body of the function contains a call to itself.

For example,

```
CS41B wants a value > 5
CS41B says > 32
```

3 To understand recursion, you must understand recursion

The Ackermann function is defined as:

$$A(m,n) = \begin{cases} n+1 & \text{if } m = 0 \\ A(m-1,1) & \text{if } m \neq 0 \text{ and } n = 0 \\ A(m-1, A(m, n-1)) & \text{otherwise} \end{cases} \tag{1}$$

Write a CS41B program that takes two values, first m and then n, and prints out the corresponding value of the Ackermann function. Place your work in a file named `ackermann.a41`.

For completeness, let us specify that $A(m,n) = 0$ when m or n is negative

For example, here are two runs of the program:

```
CS41B wants a value > -3
CS41B wants a value > 1
CS41B says > 0

CS41B wants a value > 3
CS41B wants a value > 1
CS41B says > 13
```

The function grows very quickly, and your program will make *lots* of recursive calls. Below is s a table with some values of the function. The value $A(4,2)$ requires nearly 20,000 decimal digits (see the appendix)!

	$n = 0$	$n = 1$	$n = 2$	$n = 3$	$n = 4$
$m = 0$	1	2	3	4	5
$m = 1$	2	3	4	5	6
$m = 2$	3	5	7	9	11
$m = 3$	5	13	29	61	125
$m = 4$	13	65533	...	...	...

Your program will actually only be able to produce results for only the smallest values of m and n. If you squeeze in as much stack space as you can, you ought to be able to compute $A(3, 2)$ but perhaps not $A(3, 3)$. Nevertheless, your code must be structured as an accurate representation of the recursive definition above.

4 C anyone?

This short exercise gives us a glimpse into unsafe languages, in this case, the language is the C programming language.

The program in Figure 1 at the end of this document declares a three-element array of integers, initializes all three values to 47, calls a procedure that does nothing, and then prints out the values.

Begin by creating a file `smash.c` with exactly the code in Figure 1. Then compile it with the terminal command

```
gcc -o smash smash.c
```

and run it with the command

```
./smash
```

Next, modify the program by changing the procedure `do_nothing` so that when you run the resulting program the output values are, in order, 47, 29, and 47. Make changes *only* between the braces of `do_nothing` (i.e. you're only writing the function `do_nothing`. For this exercise, we are not concerned with documentation, so you need not add comments (except for your name at the top of the file!).

The idea is to recognize that the elements of the array `a` are stored on the stack, and `do_nothing` can reach into the stack and change one of those values. You will only have to add a few lines of code, but it may take a little experimentation to find exactly the right lines. When you are finished, submit your modified version of `smash.c` in the usual way.

Here is what you need to know about the C programming language.

i. The syntax of C is very much like Java. Variables must be declared before they are used, and the equals sign is the assignment operator.

ii. The variable declarations

```
int k;
int* p;
```

declare an integer `k` and a "*pointer*" `p` to an integer. That is, `p` is a variable that holds the address in memory where an integer is stored. The expression `*p` refers to the integer *at* address `p`; it can be used on either sign of an assignment statement. Be sure that you understand the difference between the two assignment statements:

```
p = 8;
*p = 8;
```

The first sets the value of the variable `p` to be 8. The second sets the value *at the memory address stored in* `p` to be 8.

iii. It is possible to do arithmetic on addresses. For example, if `p` is the address of an integer in memory, then `p+1` is the address of the very next integer in memory. The compiler computes the right number of bytes to add to the address based on the word size. Be sure that you understand the difference between the two statements:

```
p = p + 8;
*p = *p + 8;
```

iv. Finally, the `&` operator returns the address of a variable, so that the assignment

```
p = &k;
```

stores the memory location of variable `k` in the pointer variable `p`. (More precisely, `p` receives the address in memory which has been reserved for `k`.) After that assignment, `k` and `*p` can be used interchangeably.

v. `printf` is a built-in function that prints values. It takes two parameters a string pattern and the value to be printed. There are many different patterns, but the one used in the code prints out the value as a number and followed by an end of line character. Adding some temporary print statements may help you in figuring out the solution, but make sure to remove these before you submit. Your submitted program should only print three numbers and this printing should happen with the `printf` statement already in the code.

When you're done

Make sure your code compiles. If it does not, you will not get any points for that problem.

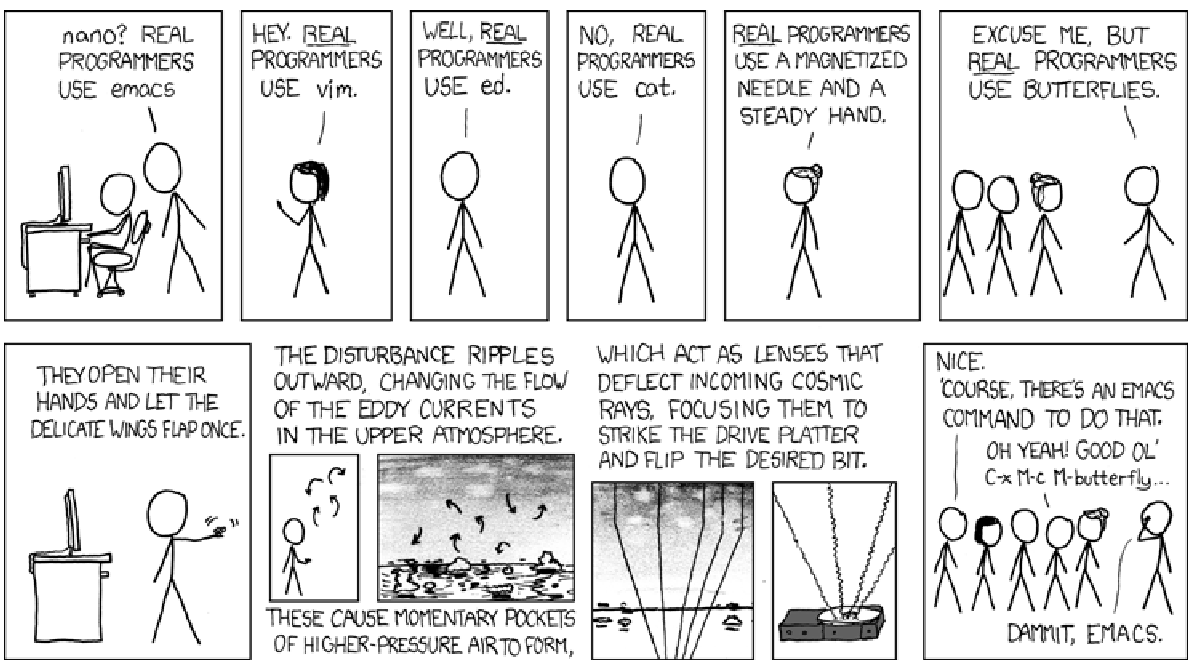
Make sure you've properly commented your code. You should include:

- A comment header at the top of the file with your name, the date, the assignment number, etc.
- Almost every line of assembly should have a comment to the right of it.

When you're ready to submit, upload your assignment via the online submission mechanism. *You should upload each of the four problems separately*, that is, you should submit four files `positive.a41`, `power2.a41`, `ackermann.a41` and `smash.c`. You may submit as many times as you'd like up until the deadline. We will only grade the most recent submissions.

Grading

`positive.a41`	4
`power2.a41`	6
`ackermann.a41`	8
`smash.c`	3
comments/style	3
Total	24



https://xkcd.com/378/

```
#include <stdio.h>

void do_nothing(){
}

int main(){
  int j;
  int a[3];

  for (j = 0; j < 3; j++){
    a[j] = 47;
  }

  do_nothing();

  for (j = 0; j < 3; j++){
    printf("%d\n", a[j]);
  }

  return 0;
}
```

Figure 1: A “do nothing” program in the C language.

Appendix

For those that are curious, Ackermann(4,2) in base 10 (all 19,729 decimal digits of it):

20035299304068464649790723515602557504478254755697514192650169737108940595563114530895061308809333481010382343429072631818229493
82118812668869506364761547029165041871916351587966347219442930927982084309104855990570159318959639524863372367203002916969592156
10876494888925409080591145703767520850020667156370236612635974714480711177481588091413574272096719015183628256061809145885269982
61414250301233911082736038437678764490432059603791244909057075603140350761625624760318637931264847037437829549756137709816046144
13308692118102485959152380195331030292162800160568670105651646750568038741529463842244845292537361442533614373729088303794601274
72495841486491593064725201515569392262818069165079638106413227530726714399815850881129262890113423778270556742108007006528396332
21550778312142885516755540733451072131124273995629827197691500548839052238043570458481979563931578535100189920000241419637068135
59840464039472194016069517690156119726982337890017641517190051133466306898140219383481435426387306539552969691388024158161859561
10064036211979610185953480278716720012260464249238511139340046435162386756707874525946467090388654774348321789701276445552940909
20219595857516229733335761595523948852975799540284719435299135437637059869289137571537400019863943324648900525431066296691652434
19174691389632476560289415199775477703138064781342309596190960654591300890188887588084733625956065444888501447335706058817090162
10849971452956834406197969056546981363116205357936979140323632849623304642106613620022017578785185740916205048971178182040018728
29399434461862243280098373237649318147898481194527130074402207656809103762039992034920239066262644919091679854615157788390603977
20759279378852241294301017458086862263369284725851403039615558564330385450688652213114813638408384778263790459607186876728509763
47127198889068047824323039471865052566097815072986114143030581692792497140916105941718535227588750447759221830115878070197553572
22414000195481020056617735897814995323252085897534635470077866904064290167638081617405504051176700936732028045493390279924918673
06539931640720492238474815280619166900933805732120816350707634351669869625020969023162859350071874190579161241536897514808261904
84794657173660100589247665544584083833479054414481768425532720731558634934760513741977952519036503219802010876473836868253102518
33775339088614261848003740080822381040764688784716475529453269476617004244610633112380211345886945322001165640763270230742924260
51582811070387018345324567635625951430032037432740780879056283663406965030844225855967039271869461158513793386475699748568670079
82396060439347885086164926030494506174341236582835214480672667684180708375486221140823657980296120002744132443843240233125740354
50193524287764308802328508558860899627744581646808578751158070147437638679769550499916439982843572904153781434388473034842619033
88841494031366139854257635577105335580206622185577060082551288893332226436281984838613239570676191409638533832374343758830859233
72228464428799624560547693242899843265267737837317328806321075321123868060467470842805116648870908477029120816110491255559832236
62448685566514026846412096949825905655192161881043412268389962830716548685255369148502995396755039549383718534059000961874894739
92880432496373165753803673586710175783994818471798498246948060532081996066183434012476096639519778021441199752546704080608499344
17825628509272652370989865153946219300460736450792621297591769829389236701517099209153156781443979124847570623780460000991829332
13068805700465914583872080880168874458355579262584651247630871485663135289341661174906175266714926721761283308452739364692445828
92571388877839056300482483799839692029222215486145902373478222682521639957440801727144146179559226175083889020074169926238300282
28624928418267124340575142418856999427233160699871298688277182061721445314257494401506613946316919762918150657974552623619122484
80638900336690743659892263495641146655030629659601997206362026035219177767406687774635493753188995878662821254697971020657472327
21372918144666659421872003474508942830911535189271114287108376159222380276605327823351661555149369375778466670145717971901227117
81278045024002638475878833939681796295069079881712169068692953824852983002347606845411417813911064856023654975422749723100761513
18700240539105109138178437217914225285874320985249578780346837033378184214440171386881242499844186181292711985333153825673218704
21530631197748535214670955334626336610864667332292409879849256691109516143618601548909740241913509623043612196128165950518666022
03071561368473236466086890501426391390651506390819937885231836505989729912540447944342516677429965981184923315155527288327402835
26884424087528112832899806259126736995462473415433335001472314306127503903073971352520693381738433229507010490618675394331307847
98015655130384758155685236218010419650255596181934986315913233036096461905990236112681196023441843363334594927631946101716652913
82371718239429921627253846177606569454229787707138319881703696458868981186321097690035573588462446483570629145305275710127887202
79653644797240254054481327483917941288264238351719491972097971459368875371987291308317380339110161285474153773777159517280841116
27597186384924222802373441925469991983672192131287035585307966942713416391033882754318613643490100943197409047331014476299861725

42442335561223743571582593338280498624389249822278071595176275784710947511903348224141202518268871372819310425347819612844017647
95315050571107229743145699152234516431218486575757865281975648435089583847229235345594645212158316577514712987082259092926556388
36651120681943836904116252668710044560243704200663709001941185557160472044643696932850060046928140507119069261393993902735534545
56747031490388602202463994826050176243196930564066636662609020704888743889890749815286544438186291738290105182086993638266186830
39152732645812867828066013375000965933646251460917231803129303478774212346791184547913111098977946482169225056293999567934838016
99157439700537542134485874586856047286751065423341893839099110586465595113646061055156838541217459801807133163612573079611168343
86376766730735458349478978831633012924080083635682593915711313097803051644171668251834657367593419808495894794098329250008638977
85634946932124734261030627137450772861569225966285738579055332406418490184513282846327092697538308673084091422476594744399733481
30810986399417379789657010687026734161967196591599588537834822988270125605842365589539690306474965584147981310997157542043256395
77607048510088157829140825077773855979012912940730946278594450585941227319481275322515232480150346651904822896140664689030510251
09162377704484862302294889667113805556079566207324493733740278367673002030116152270089218435156521213792157482068593569207902145
02277133099987729459596952817044582181956080965811702798062669891205061560742325686842271306295009864421853470810407128917646906
55083612991669477802382250278966784348919940965736170458678624255400694251669397929262471452494540885842272615375526007190433632
91963757775021760051958006938476357895868784895368721228985578068265181927036320994801558744555751753127364714212955364940843855
86615208012115079075068553344489258693283859653013272046970694571546959353658571788894862333292465202735853188533370948455403336
56535698817258252891805663548836374379334841184558016833182767683464629199560551347003914787680864032262961664156066750815371064
67231084619642475374905537448053182260027102164009805844975260230356400380834720531499411729657367850664214008426964971032419191
82121213206939769143923368374709228267738708132236680086924703491586840991153098315412063566123187504305467536983230827966457417
62080659317726568584168183796610614496343254411170694170022265781735835125982108076910196105222926387974504901925431190062056190
65774524161919131875339840493439768233102984658933183730158095925228292068208622303325852801192664963144413164427730032377922747
12330696417149945532261035475145631290668854345426869788447742981777493710117614651624183616680254815296335308490849943006763654
80610294009469375060984558855804397048591444958444507997849704558355068540874516331646411808312307970438984919050658758642581073
84224205911919416741824904527002882639830579500573417114870311871428341844991534567029152801044851451760553069714417613685823841
02787659324662689978418319620312262421177391477208004883578333569204533935953254564897028558589735505751235129536540502842081022
78524877660357424636667314868027948605244578267362623085297826505711462484659591421027812278894144816399497388188462276824485162
20518170767221698632657016543169197426512300417573299044735376725368457927543654128265535818580468400693677186050200705472475484
00805530424951854495267247261347318174742180078574693465447136036975884118029408039616746946288540679172138601225419503819704538
41726800639882065632879283958270851091995883944829777564715202613287108952616341770715164289948795356485455355314875497813400996
48544986358248476905900331169613037661279234643231297066284113074270462020320133683503854253603136367635752126047074253112092334
02837482949453104727418969287275572027615272268283376741393425652653283068469997597097750005560889932685025049212884068274139881
63154045649035077587168007405568572402175868543905322813377070741583075626962831695568742406052772648585305061135638485196591896
86495963355682169754376214307786659347304501648224329648912707098980766766256715172690620588155496663825738292741820822789606844
88222983394816670984039024283514306813767253460126007269262969468672750794346190439996618979611928750519442356402644303271737341
59128149605616835398818856948404534231142461355992527233006488162746672352375123431189344211888508507935816384899448754475633168
92138696755743027379537852625423290248810471819390372206668947022042588368958409399984535609488699468338525796751618821594109816
24918741813364726965123980677561947912557957446471427868624053750576104204267149366084980238274680575982591331006919941904651906
53117190892607794911921794640735512963386452303567334558803331319708036545718479155043265489955970586288828686660661802188224860
21449999731221641381706534801755104384066244128228036166489042573776409563264828252584076690456084394903252905263375323165090876
81336614242398309530806549661879381949120033919489494065132398816642080088395554942237096734840072642705701165089075196155370186
26479745638118785617545711340047381076276301495330973517418065547911266093803431137853253288353335202493436597912934128485497094
68263290758301930726653377825593143311109638480539408592839889077962104798479196868765399874770959127887274758744398067798249682
78272200926449944559380414608770641941810440758269805688038949654616587983904660587645341810289907194293021774519976104495043196
84150345551404482092893337865736305283061999007774872692299860827905317169187657886090894181705799340489021844155979109267686279
65975839524839267348836347456516870161662406424242412289611180106156823425393921800524834547237792199112285959141918774917938233
40010078128326506710281781396029120914720100947878752551263372884222353869490067927664511634758101193875319657242121476038284774

77457170457861041738574791130190858387789015233434301300528279703858035981518292960030568261209195094373732545417105638388704752
89505639610298436413609356416325894081379815116933386197973398216707610046079800960160248230969430438069566201232136501405495862
50615282588033022908385812478469315720323233601899469437647726721879376826431828382603564520699468630216048874528424363593558622
33350623594500289055858161127534178375045593612613085264082805121387317749020024955273873458595640516083058305377073253397155262
04447054295735383611136775231699727402929416742044232481138750756313190782721888640533746942138421699288629404796353051505607881
26366206497231257579019598873041195626227343728900516561111094111745277965482790471250581999077498063821559376885546498822938985
40829132512907647838632249478101675349169348928810420301561028338614382737816094634133538357834076531432141715065587754782025245
47806573013422774706167442419689526131642741046954746214837562882997718041867850845469656191509086958742511844358373065909514609
80451247409411373899927822492983367796011015387096129749705566301637307202750734759922943792393824427421186158236161317886392553
09511718842129850830723825972914414225157940388301135908333165185823496722125962181250705811375949552502274727467436988713192667
07692991990844671612287388584575846227265733307537355728239516169641751986750126817454293237382941438248143771398619067166575729
45807804820559511881687188075212971832636442155336787751274766940790117057509819575084563565217389544179875074523854455200133572
03333237989507439390531291821225525983379090946363020218535384885482506289771561696386071238277172562131346054940177041358173193
17633701363322528191275471914434509207118488383668181742633429496118700915030491653394647637177664391207983474946273978221715020
90670190302469762151278521956142070806461631373236517853976292092025500288962012970141379640038055734949269073535145961208674796
54773369295877362863566014376796403843079686413856344780132826128458918489852804804884418082163942397401436290348166545811445436
64600324906187630395023564020445307482102413668951966442213392007574791286838051751506346625693919377402835120756662608298904918
77287833852178522792045771846965855278790447562192663992008409302075673925363735628390829817577902153202106409617373283598494066
65214119818381088451545977289516457213189779790749194101314836854463961690460703010759681893374121757598816512700076126278916951
04063158576375347874200702220510708912576123616580268068158584998526314658780866168007332646768302063916972030648944056281954061
90685242003053463156621891327309069687353181641094514288036605995220248248886711554429104721929134248346438705368508648749099178
81267056566538719104972182004237149274016446094345984539253670613221061653308566202118896823400575267548610147699368873820958455
22115719234796868881608536316158628801503959494185294892270744108282071693033878180849362040182552222710109856534448172074707560
19245915599431072949578197878590578940052540122867517142511184356437184053563024181225473266093302710397968091064939272722683035
41046763259135527968383770501985523462122285841055711992173171796980433931770775075562705604783177984444763756025463703336924711
42208155199736913719751632413027487121998634045482485245701185533426752647159783107312456634298052214554941562527240289153333543
49341217862037007260315279870771872491234494477147909520734761385425485311552773301030342476835865496093722324007154518129732692
08105842409055772564580368146223449318970813889714329983134761779967971245378231070373915147387869211918756670031932128189680332
26965944592862106074388274169194651622676325406650708810710303941788605648937698167341590259251946118236429456526693722031555047
00213598846292758012527715422016629954863130324912311029627923723899766416803497141226527931907636326136814145516376656559839788
48938173308266877990196288693229659737995193162118721545528739417024366988559388879331674453336311954151840408828381519342123412
28200309503133410507047601599879854725291906652224793197154403317948368373732208218857733416238564413807005419135302459439135025
54531886454796252260251762928374330465102361057583514550739443339610216229675461415781127197001738611494279501411253280621254775
81051297208846526315809480663368767014731073354071771087661593585681409821296773075919738297344144525668877085532457088895832099
38234321027182241147637327913575686154212528496579033350931527769255058456440105521926445053120737562877449981636463328358161403
30175813967359427327690448920361880386754955751806890058532927201493923500525845146706982628548257883267398735220457228239290207
14482221988558710289699193587307427781515975762076402395124386020203259659625021257834995771008562638611823381331850901468657706
40106762786175837727728958927460394039303372718738505369129571267150668966884938808851429436099620129667590792250822753138128498
51526902931700263136328942095797577959327635531162066753488651317323872438748063513314512644889967589828812925480076425186586490
24111112730135719718138160258317850693224400799865663537154408845486639318170839573578079905973083909488180406093595919090747396
09044101505163217496814121007657191774837673557510007336169223865374290794578032000423374528075661530429290144957806296341383835
51783599764708851349004856973697965238695845994595592090709058956891451141412684505462117945026611750166928260250950770778211950
43261738322356243760177679936279609936897519139496503335850715541843645685261667424368892037103749532842592713161053783498074073
91586338179676584252580367372064693512486522384813416638080615057048290598906964519364400185971204257230073164100099169875242603
77362177763430621616744884930810929901009517974541564251204822086714586849255132444266777127863728211331536224301091824391243380

2140462422223349153559516890816288487989988273630445372432174280215755777967021666317047969728172483392841015642274507271779269399929740308072770395013581545142494049026536105825409373114653104943382484379718606937214444600826798002471229489405761853892203425608302697052876621377373594394224114707074072902725461307358541745691419446487624357682397065703184168467540733466346293673983620004041400714054277632480132742202685393698869787607009590048684650626771363070979821006557285101306601010780633743344773073478653881742681230743766066643312775356466578603715192922768440458273283243808212841218776132042460464900801054731426749260826922155637405486241717031027919996942645620955619816454547662045022411449404749349832206807191352767986747813458203859570413466177937228534940031631599544093684089572533438702986717829770373332806801764639502090023941931499115009105276821119510999063166150311585582835582607179410052528583611369961303442790173811787412061288182062023263849861515656451230047792967563618345768105043341769543067538041113928553792529241347339481050532025708728186307291158911335942014761872664291564036371927602306283840650425441742335464549987055318726887926424102147363698625463747159744354943443899730051742525110877357886390946812096673428152585919924857640488055071329814299359911463239919113959926752576359007446572810191805841807342227734721397723218231771716916400108826112549093361186780575722391018186168549108500885272274374212086524852372456248697662245384819298671129452945515497030585919307198497105414181636968976131126744027009648667545934567059936995464500558921628047976365686133316563907395703272034389175415267500915011198856872708848195531676931681272892143031376818016445477367518353497857924276463354162433601125960252109501612264110346083465648235597934274056868849224458745493776752120324703803035491157544831295275891939893680876327685438769557694881422844311998595700727521393176837831770339130423060958999137314684569010422095161967070506420256733873446115655276175992727151877660010238944760539789516945708802728736225121076224091810066700883474737605156285533943565843756271241244457651663064085939507947550920463932245202535463634444791755661725962187199279186575490857852950012840229035061514937310107009446151011613712423761426722541732055959202782129325725947146417224977321316381845326555279604270541871496236585252458648933254145062642337885651464670604298564781968461593663288954299780722542264790400616019751975007460545150060291806638271497016110987951336633771378434416194053121445291855180136575558667615019373029691932076120000925506508158327550849934076879725236998702356793102680413674571895664143185267905471716996299036301554564509004480278905570196832831363071899769915316667920895876857229060091547291963638167359667395997571032601557192023734858052112811745861006515259888384311451189488055212914577569914657753004138471712457796504817585639507289533753975582208777750607233944558789590571915673