In [1]:
```
from sympy.ntheory import factorint

N = 12
display(factorint(N))
```

```
{2: 2, 3: 1}
```

In [2]:
```
N = 340282366920938463463374074317682310984309834 3
display(factorint(N))
```

```
{3: 2, 74519450661011221: 1, 507372928070793263124358078 7: 1}
```

In [3]:
```
RSA1024 = 135066410865995223349603216278805969938881475605667027 5
display(RSA1024)
```

```
135066410865995223349603216278805969938881475605667027524485143851 5
265106048595338339402871505719094417982072821644715513736804197039 6
419174304649658927425623934102086438320211037295872576235850964311 0
564073501508187510676594629205563685529475213500852879416377328533 9
061097505443349998111500569772368909275 63
```

In [4]:
```
RSA250 = 214032465024074496126442307283933356300861471514475501 77

p = 641352894770715802787901901705773890848250147429434472081168 5
q = 333720275949781565562260106053551142279407603447675546667845 2

display(RSA250 == p * q)
```

```
True
```

In [1]:
```
import math

N = 463675969018391834968223957323668663263635331975581842139366 7
M = 505671487480487786422516484397774937475102137917608354042646 1

display(math.gcd(N, M))
```

```
135066410865995223349603216278805969938881475605667027524485143851 5
265106048595338339402871505719094417982072821644715513736804197039 6
419174304649658927425623934102086438320211037295872576235850964311 0
564073501508187510676594629205563685529475213500852879416377328533 9
061097505443349998111500569772368909275 63
```

In [2]:
```
display(math.gcd(N**200 + 1, M**100 + M**2))
```

```
113
```

In [3]:
```
K = 641352894770715802787901901705773890848250147429434472081168
N = 463675969018391834968223957323668663263635331975581842139366
M = 505671487480487786422516484397774937475102137917608354042646
display(pow(N, K**10 + 1, M**10 + 1))
```

```
542080890895977406028898753671977045621731891209489822571389293690
049029205875268391016773496273016272919329382669533127141405381618
199558711478111330716831139767411065605643486113628392097489109741
279989438546490061766446832927171765506749534185882270482964957955
724483155264208314048742469387847020407987537873027379036515942355
064255761049423944523900400810690418585252179850447120180289913946
545887022535847609923204071461037767259867972001546270283393458045
151626243131178465271884695260294198503010478207039551370657556801
805463289396833897828931019379770231600270208246121183509811722999
778486236491442678439570530462594698162087656590363032142671356460
636615003526450559450988257054265576673783996389684225946503950767
649117659125004789711204077120438327267065399992029357329368204005
497085533829465336908163404880411532025618221114554409481610762912
112742249226319353536843143250675559791743355892993051078488285292
271780234384025830548398321095037814715965000729591077070200657825
586132525389117446847982845507070180002421259314497140083214078606
620665224339027746048978496589265602430286475229240500730395030244
673737607732257240901560040493525946358670279354659624147644563735
529095699930579496461756154827056521608042123837065251899742149917
722058617256564053290758153841574743341020230723665105568383923335
249812623953148372922913512135798532338006645439596975706689689325
510539120089434512240390112685580833929289214461803464764059412963
891809347433516954054255735374114229687467253661509404329081038467
369764853140054577946427932407289077167481135826230478969272995253
480346178433989934901133184236611983498525476702279781754198284213
038797346045524543446837609691961446346898034481960162369906547207
517407033255952737243135098667981397918627389414124350884907904739
254265525618735905264899522657981378010176370063278417731899859396
123977376065188642849714402071821332656099246595192273548124579308
543693791667913755043661779633671673577342042501345239010120537284
450566208371948327714999829734560085020796859461161050205811975405
458059616153460321784141539990056380808571117172677389938997159425
769490650371331381972471611151276595851094700767842128466898434498
429452828882911743581872753258319074174268660109058053405028959201
761135679603364904072305569448382741870366244397396582565453282641
021954070239815306917514968982226339421982302709070989132539956519
861154506619395018871824565319783779159642632956094134775393965042
611447245577078846931722372616367550006355620360276455568080512099
351791754167561432339509342147366219500399025066442214439984353414
045247646617141997255839564468997146574790767489679082209766030710
962931957694684033978412952567599593488687529011526359106169809555
796674018667232779312015101776303021338479321582789903126823345493
765252062693583573954771298331432079857389422907643719070485636439
327795151484278956526648597260084828697308427169704251546673832583
097054565752815337221780291031922567653435046451128269020739602234
798962448167266870397240841466589440334718167916077421753615336442
436840721557202572348021932511877708639899866042064925150997870115
101300919233087163724915056002346383285602529985612554358728670791
489881983889831150243019029551221559467757627524822095467570419971
880196887650126401810574270118295619216693001247689245264372564423
558304210872792042648342762907900896926369841981769307645977885296
041082118852278201677791079670048344982 6
```

In [1]:

```
1 import galois
2
3 GF = galois.GF(2)
4
5 N, n = 1000, 1000
6
7 A = GF.Random((N, n))
8 B = A.null_space()
9 display(B)
```

```
---------------------------------------------------------------------
--------
ModuleNotFoundError                       Traceback (most recent ca
ll last)
Cell In[1], line 1
----> 1 import galois
      3 GF = galois.GF(2)
      5 N, n = 1000, 1000

ModuleNotFoundError: No module named 'galois'
```

In [2]:

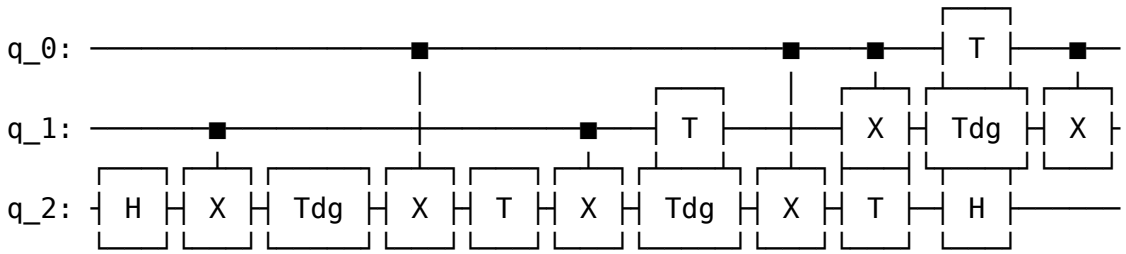
```
from qiskit import QuantumCircuit
from qiskit.quantum_info import Statevector, Operator
from qiskit.visualization import array_to_latex

Toffoli = QuantumCircuit(3)

Toffoli.h(2)
Toffoli.cx(1, 2)
Toffoli.tdg(2)
Toffoli.cx(0, 2)
Toffoli.t(2)
Toffoli.cx(1, 2)
Toffoli.tdg(2)
Toffoli.cx(0, 2)
Toffoli.t(1)
Toffoli.t(2)
Toffoli.cx(0, 1)
Toffoli.t(0)
Toffoli.tdg(1)
Toffoli.h(2)
Toffoli.cx(0, 1)

display(Toffoli.draw())
U = Operator(Toffoli)

display(array_to_latex(U))
```



$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$