#Cicada
#RSA N=P*Q
In PyCharm or Python

Abstract

The purpose of this paper is to provide algorithm that is 4 lines of code and that finds P & Q when N is given. It will work for RSA-1024 & RSA-2018 if the computer can float large numbers in PyCharm or Python.

I. RSA-100

Given N below (A 100 digits). To find P and Q. First solve for P by multiplying A*A and then dividing it by A+A. Then continue by dividing by the square root of A that is divided by 2 and 1E48 is added to it. To solve for Q divide A by P. The algorithm will solve for P & Q and print (N,P,Q).

```
import math

#RSA 100

A=152260502792253336053561837813263742971806811496138068865790849458012296325895289765400035069200613 9
P=((A*A)/(A+A))/((math.sqrt(A)/2)+10000000000000000000000000000000000000000000000000) #49 digits 99+1/2=49
Q=(A/P)
N=P*Q print
(N,P,Q)
```

P=3.7118083904089913e+49 Q=4.102057185540126e+49 N=1.5226050279225333e+99
#Actual P=37975227936943673922808872755445627854565536638199 × Q=4009469095092088103068373529276146838921489972406 1

II. RSA-110

Given N below (A 110 digits). To find P and Q. First solve for P by multiplying A*A and then dividing it by A+A. Then continue by dividing by the square root of A that is divided by 2 and 1E53 is added to it. To solve for Q divide A by P. The algorithm will solve for P & Q and print (N,P,Q).

```
import math

#RSA 110

A=35794234179725868774991807832568455403003778024228226193532908190484670252364677411513516111204504060317568667
P=((A*A)/(A+A))/((math.sqrt(A)/2)+1000000000000000000000000000000000000000000000000000000) #55 digits 110+1/2=55
Q=(A/P)
N=P*Q print
(N,P,Q)

#N=3.5794234179725868e+109 P=5.7892978070978474e+54 Q=6.182828275968304e+54
#Actual= Q=6122421090493547576937037317561418841225758554253106999 × P=5846418214406154678836553182979162384198610505601062333
```

III. RSA-2048

Given N below (A 617digits). To find P and Q. First solve for P by multiplying A*A and then dividing it by A+A. Then continue by dividing by the square root of A that is divided by 2 and 1E307 is added to it. To solve for Q divide A by P. The algorithm will solve for P & Q and print (N,P,Q).

```
import math

#RSA-2048

A=25195908475657893494027183240048398571429282126204032027777137836043662020707595556264018525880784406918290641249515082189298559149
17618450280848912007284499268739280728777673597141834727026189637501497182469116507761337985909570009733045974880842840179742910064245
86918171951187461215151726546322822168699875491824224336372590851418654620435767984233871847744479207399342365848238242811981638150
10674810451660377306056201619676256133844143603833904414952634432190114657544454178424020924616515723350778707749817125772467962926386
356373289912154831438167899885040445364023527381951378636564391212010397122822120720357
P=((A*A)/(A+A))/((math.sqrt(A)/2)+10000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000) #617 digits/2=308
Q=(A/P)
N=P*Q print
(N,P,Q)

OverflowError: integer division result too large for a float P.
```

IV. Conclusion

If your computer can process P, one will get N=P*Q for RSA-2048. This algorithm is 4 lines long and can find P & Q when N is given. The algorithm does have a margin of error but given N, the algorithm will return the aproximate P & Q.