RSA Encryption/Decryption

Pick large primes

```
In[19]:= {p = RandomPrime[10^20], q = RandomPrime[10^20]}
Out[19]= {72 138 082 000 727 296 459, 57 548 602 841 590 543 873}
```

Compute their product, which is the modulus for the public key

```
In[20]:= n = p q
Out[20]= 4 151 445 830 813 946 559 402 157 726 486 717 045 707
```

Pick a random number relatively prime to $(p-1)(q-1)$. This is the (public) encryption exponent.

```
In[21]:= While[GCD[e = RandomInteger[n], (p - 1) (q - 1)] ≠ 1]; e
Out[21]= 699 181 861 313 198 203 642 760 806 807 082 745 893
```

Find the inverse of e mod $(p-1)(q-1)$. This is the (private) decryption exponent.

```
In[22]:= d = PowerMod[e, -1, (p - 1) (q - 1)]
Out[22]= 1 094 470 713 330 792 831 223 718 655 002 093 756 333
```

Make up a message and convert the characters to numerical codes.

```
In[23]:= mess = ToCharacterCode["Your Message Here"]
Out[23]= {89, 111, 117, 114, 32, 77, 101, 115, 115, 97, 103, 101, 32, 72, 101, 114, 101}
```

Then use these as the digits of a number in base 128.

```
In[24]:= x = FromDigits[mess, 128]
Out[24]= 466 654 477 002 722 201 989 670 496 462 535 013
```

Encrypt.

```
In[25]:= y = PowerMod[x, e, n]
Out[25]= 2 193 539 735 121 106 584 975 809 322 714 998 115 973
```

Converting this back to characters looks like garbage, of course.

```
In[26]:= FromCharacterCode[IntegerDigits[y, 128]]
Out[26]= d<14+}   |Zz &gh
```

Decrypt.

```
In[27]:= z = PowerMod[y, d, n]
Out[27]= 466 654 477 002 722 201 989 670 496 462 535 013
```

Convert back to characters and recover the original message.

```
In[28]:= FromCharacterCode[IntegerDigits[z, 128]]
Out[28]= Your Message Here
```

Pollard's Factoring Algorithm

(First we'll do a baby example)

```
In[38]:= n = RandomPrime[10^3] * RandomPrime[10^3]
Out[38]= 87 287
```

```
In[39]:= f[x_] := Mod[x^2 + 2, n]
```

```
In[40]:= y = z = y0 = RandomInteger[n]
Out[40]= 27 071
```

(We should evaluate this next line repeatedly until we see a factor)

```
In[49]:= {y = f[y], z = f[f[z]], GCD[y - z, n]}
Out[49]= {45 025, 10 836, 191}
```

...

Now we'll get serious!

```
In[33]:= n = RandomPrime[10^5] * RandomPrime[10^5]
Out[33]= 1 971 164 003
```

In[34]:=
```
y = z = y0 = RandomInteger[n]
```

Out[34]= 723 660 521

Repeat 100 times and hope it’s enough to see a factor. If not, we should repeat some more.

In[37]:=
```
Table[{y = f[y], z = f[f[z]], GCD[y - z, n]}, {100}] // TableForm
```

Out[37]//TableForm=

1 324 078 819	1 832 925 274	1
792 238 892	1 043 071 445	1
242 407 378	1 423 195 149	1
1 676 503 461	715 289 188	1
416 575 284	689 217 703	1
82 698 264	828 170 498	1
81 105 123	1 722 529 844	1
591 693 726	79 293 116	1
1 078 188 473	1 828 411 512	1
59 307 191	1 166 568 306	1
1 799 685 289	759 903 136	1
1 926 321 133	1 845 082 003	1
32 176 452	354 925 020	1
1 709 356 604	322 173 264	1
1 757 258 467	1 241 987 333	1
931 385 900	1 430 161 903	1
227 867 026	1 923 695 946	1
434 012 232	1 001 460 001	1
1 349 226 502	1 048 253 548	1
220 843 785	504 694 851	1
632 342 040	1 230 104 371	1
926 198 719	1 207 725 865	1
1 755 178 869	189 859 360	1

958 179 943	315 616 336	1
1 741 493 641	720 353 608	1
306 434 854	698 818 295	1
1 086 431 306	712 744 938	1
1 314 433 674	1 556 155 576	1
285 842 093	1 888 660 075	1
162 893 199	1 246 473 541	1
826 550 063	602 852 469	1
233 887 075	128 603 192	1
1 934 180 512	1 457 953 537	1
1 703 013 404	1 748 108 017	1
354 799 855	1 871 075 758	1
296 058 325	1 847 397 367	1
259 758 481	1 081 092 173	1
918 963 044	1 536 686 485	1
382 603 294	1 559 905 991	1
865 344 331	952 949 236	1
1 579 838 691	603 641 074	1
507 941 787	279 599 902	1
949 504 586	1 480 513 698	1
1 099 129 713	1 516 356 257	1
1 124 147 694	1 843 388 919	1
1 446 349 561	458 634 165	1
376 174 200	1 357 332 065	1
1 475 270 319	404 744 814	1
1 079 929 933	718 818 892	1
457 908 067	421 311 766	1
432 303 712	1 967 375 313	1
714 976 277	1 601 483 310	1
1 112 440 859	729 660 334	1
1 862 795 418	1 595 267 037	1

568 092 905	1 604 443 972	1
874 138 929	300 809 998	1
1 589 285 396	637 829 350	1
211 145 515	722 498 887	1
1 187 923 297	731 508 468	1
1 510 345 919	8 661 584	1
1 672 313 352	575 188 166	1
1 309 188 437	780 389 310	1
366 703 040	722 918 457	1
548 132 173	23 908 815	1
1 142 595 751	1 739 818 004	1
1 377 942 801	957 732 480	1
1 491 703 592	5 692 699	1
861 873 102	951 556 274	1
1 672 860 448	1 002 328 298	1
1 353 559 881	592 284 696	1
749 808 641	447 878 193	1
89 198 534	1 007 940 923	1
1 653 500 003	1 261 254 774	1
1 116 088 063	290 895 461	1
1 840 095 303	60 907 235	1
360 796 531	545 925 273	1
1 560 272 300	747 137 799	1
978 047 481	886 210 269	1
1 939 833 526	831 650 396	1
509 997 594	879 876 516	1
1 590 755 103	1 833 566 989	1
1 554 660 161	1 034 776 446	1
569 685 154	1 561 011 595	1
977 550 428	1 618 454 602	1
323 678 730	1 552 310 225	1

1 569 042 062	446 691 883	1
1 555 946 113	1 187 394 712	1
1 904 168 093	774 476 954	1
1 136 712 934	1 670 347 986	1
1 896 768 524	223 891 419	1
1 757 751 965	780 661 771	1
1 601 784 997	1 713 561 438	20 983
109 668 790	1 605 582 236	1
1 046 343 320	133 716 701	1
1 362 458 526	1 620 344 062	1
985 647 484	1 502 711 604	1
838 420 803	1 248 300 305	1
805 091 563	810 688 144	1
206 449 242	17 692 321	1
868 563 378	1 841 592 625	1