# COM 5335 ASSIGNMENT #1

## Objective

Implement the Miller-Rabin primality test and the Rabin Public-Key Cryptosystem.

## Description

You need to deal with big number arithmetic. You can use software packages/classes for this part or implement in your own way.

There are 2 things you need to do before implementing the Rabin public-key cryptosystem.

(1) Miller-Rabin primality test
(2) 256-bit prime number generation

Prime number generation involves pseudorandom number generation. Here you can use any pseudorandom number generating functions provided in any library (such as `srand()` in `<cstdlib>`) to do this assignment. Remember, in practice, pseudorandom number generators need to be cryptographically secure and `srand()` is NOT a good choice.

Miller-Rabin as well as any other primality tests are EXPENSIVE, so it is better to avoid running it as much as possible. You can significantly reduce your program's running time if you DO TRIAL DIVISION FOR SMALL PRIMES. IT IS STRONGLY RECOMMENDED. Below is a list of small primes up to 1000.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 3 | 5 | 7 | 11 | 13 | 17 | 19 | 23 | 29 | 31 |
| 37 | 41 | 43 | 47 | 53 | 59 | 61 | 67 | 71 | 73 | 79 |
| 83 | 89 | 97 | 101 | 103 | 107 | 109 | 113 | 127 | 131 | 137 |
| 139 | 149 | 151 | 157 | 163 | 167 | 173 | 179 | 181 | 191 | 193 |
| 197 | 199 | 211 | 223 | 227 | 229 | 233 | 239 | 241 | 251 | 257 |
| 263 | 269 | 271 | 277 | 281 | 283 | 293 | 307 | 311 | 313 | 317 |
| 331 | 337 | 347 | 349 | 353 | 359 | 367 | 373 | 379 | 383 | 389 |
| 397 | 401 | 409 | 419 | 421 | 431 | 433 | 439 | 443 | 449 | 457 |
| 461 | 463 | 467 | 479 | 487 | 491 | 499 | 503 | 509 | 521 | 523 |
| 541 | 547 | 557 | 563 | 569 | 571 | 577 | 587 | 593 | 599 | 601 |
| 607 | 613 | 617 | 619 | 631 | 641 | 643 | 647 | 653 | 659 | 661 |
| 673 | 677 | 683 | 691 | 701 | 709 | 719 | 727 | 733 | 739 | 743 |
| 751 | 757 | 761 | 769 | 773 | 787 | 797 | 809 | 811 | 821 | 823 |
| 827 | 829 | 839 | 853 | 857 | 859 | 863 | 877 | 881 | 883 | 887 |
| 907 | 911 | 919 | 929 | 937 | 941 | 947 | 953 | 967 | 971 | 977 |
| 983 | 991 | 997 | | | | | | | | |

Implement Rabin Cryptosystem. Users are asked to input two 128-bit primes p and q, and a 96-bit plaintext in hex. Do 16-bit repetition padding at the end as described in the class. For decryption, users are asked to input the ciphertext as well as p and q. Sample I/O is shown below.

output your 256-bit prime number

## Sample I/O (`Input shown in` **`bold face`**.)

```
<Miller-Rabin>
xxxxxxxx  xxxxxxxx  xxxxxxxx  xxxxxxxx  xxxxxxxx  xxxxxxxx  xxxxxxxx
xxxxxxxx

<Rabin Encryption>
p = daaefe65 2cad1614 f17e87f2 cd80973f
q = f9998862 6723eef2 a54ed484 dfa735c7
n = pq = d5375c87 792a4ac9 135966b6 d1689939 c249ed22 452f77d6
3fa82d67 e95e9cf9

Plaintext: be000bad bebadbad bad00deb deadface deafbeef add00add
bed00bed
Ciphertext = 205651dd a3fced3e 74e9c50a 61342e29 b6b8e14e 85ce5666
```

7b341c78 cc2965cb

<Rabin Decryption>
Ciphertext = **5452361a db4c34be 04a5903a e00793bc 1086e887 ebed06e2 3ffba0b4 a4348cc0**
Private Key:
p = **d5e68b2b 5855059a d1a80dd6 c5dc03eb**
q = **c96c6afc 57ce0f53 396d3b32 049fe2d3**
Plaintext = 00000000 12345678 87654321 12345678 87654321 12345678 87654321