Fast exponentiation mod n

- Algorithm calculates a^e mod n so, that largest number stored in variables ⊙ $(n-1)^2$. Algorithm is fast, and overflow is avoided.

Algorithm:

Four BigInteger variables are needed:

A = base, E = exponent

n = modulus , C = additional varíable

1. Set C = 1
2. While E > 1 DO

   If E is odd,

   then {E = E-1 ; C = A*C mod n ;}

   else { E = E/2 ; A = A^2 mod n ; }

3. A = A*C mod n

Return A

Example 7^{11} mod 13

7^{11} mod 13 =

$7^{10} * 7$ mod 13 =

$49^{5} * 7$ mod 13 =

$10^{5} * 7$ mod 13 =

$10^{4} * 5$ mod 13 =

$100^{2} * 5$ mod 13 =

$9^{2} * 5$ mod 13 =

81 *5 mod 13 =

3 *5 mod 13 =

2

Variables:

A	E	C	N
7	11	1	13
7	10	7	13
49=10	5	7	13
10	4	7*10=5	13
100=9	2	5	13
81=3	1	5	13
3*5=**2**	1	1	13

result