

Compute(x, b, n)
// Return $(x^b \bmod n)$ by repeated squaring, the algorithm is linear in the length of n.

```
if b == 0 then
    return 1;

y = 1;
while b > 0
    k =  $\lfloor \log_2 b \rfloor$ ;
    p = x;
    for i = 1 to k
        p = p2 mod n;
    end for

    y = y*p mod n;
    b = b - 2k;
end while

return y;
end
```

Iterative implementation in Matlab:

function y = compute_i(x, b, n)

```
if b == 0
    y = 1;
    return;
end

y = 1;

while b > 0
    k = floor(log2(b));
    p = x;
    for i = 1:k
        p = mod(p*p, n);
    end

    y = mod(y*p, n);
    b = b - (2k);
end
```

Recursive implementation in Matlab:

function y = compute(x, b, n)

```
if b == 1
    y = x;
```

```
    return;
end

if mod(b, 2) == 0
    z = mod(compute(x, b/2, n), n);
    y = mod(z*z, n);
else
    z = mod(compute(x, b-1, n), n);
    y = mod(x*z, n);
end
```

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.