**Algorithm:** Successive Squaring

**Input:** A base, power, and a modulus

**Output:** A to the power of B mod M

```
1: Initialize highestPower ← 0
2: for i = 0 to 20 do
3:    if powerOf(2, i) > B then
4:       highestPower ← i
5:    end if
6: end for
7: for i = highestPower − 1 to 0 do
8:    if powerOf(2, i) <= B then
9:       B ← B − powerOf(2, i)
10:      D[i] ← true
11:   else
12:      D[i] ← false
13:   end if
14: end for
15: M[0] ← A
16: for i = 0 to |M| do
17:   M[i] ← powerOf(M[i − 1], 2) % n
18: end for
19: Initialize total ← 1
20: for i = 0 to |D| do
21:   if D[i] then
22:      total ← total ∗ A[i]
23:      total ← total % n
24:   end if
25: end for
26: return total
```