Exercises for tomorrow

```
fmod NAT−ADD is
   sort Nat .
   op 0 : → Nat [ctor] .
   op s : Nat → Nat [ctor] .
   op _+_ : Nat Nat → Nat .

   vars M N : Nat .

   eq 0 +M =M .
   eq s(M) +N =s(M +N) .
endfm
```

Make a new module which extends NAT−ADD and

1. define a function **op** double : Nat → Nat . which doubles its argument. For example, double(0) should be 0 while double(s(s(s(0)))) should be s(s(s(s(s(s(0)))))). Do not use +, only 0 and s.
2. define a function **op** half : Nat → Nat . which divides a number by 2. For example, “half” of 0 is 0; “half” of 2 is 1; “half” of 3 is also 1; “half” av 4 is 2; “half” of 5 is 2. What is half of 86? of 87?
3. define a function **op** monus : Nat Nat → Nat which computes “minus down to 0,” i.e., max(m− n,0).
4. define a function **op** diff : Nat Nat → Nat . which computes the difference between two numbers. For example, the diff between 2 and 7 is 5 and the diff between 8 and 1 is 7.