Consider the computation:

```
int f(int a, int b, int c, int d, int x, int y, int z)
{
   return(a*x*x*x+b*y*y+c*z+d);
}
```

Assume:

- Multiplier (32b): Area=10, Delay=10
- Adder (32b): Area=1, Delay=1
- Multiplexer(32b): Area=1, Delay=0.1
- Register(32b): Area=0.5, Delay=0 (don't charge any extra delay)

1. How many multiplies in the computation? ☐
2. How many adds in the computation? ☐
3. Assuming a direct spatial implementation (like Day 15):
   (a) What is the area? ☐
   (b) What is the delay? ☐
4. How would you implement this design using only a single multiplier, a single adder along with multiplexers and registers (the number of which you pick, but should try to minimize)?
   (a) What is the area? ☐
   (b) What is the delay? ☐
5. How would you implement this design so that it had a delay within 10% of the first (fully spatial) case, but used less area? Within 10% is more tricky than intended; Maybe only 60% slower is the easy target.
   (a) What is the area? ☐
   (b) What is the delay? ☐