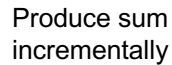
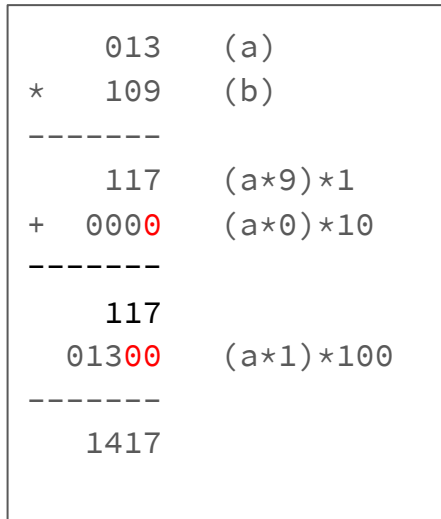
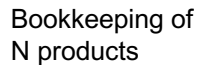


Multiplication

- Consider: $109 \times 13 = 1417$
- Approach: Successive Additions
 - Consider: $9 + 9 + 9 \dots + 9$ (13 times) = ?
 - What is carry value for the 10's column?
- Approach: Long Multiplication



x13

Algorithm for Decimal Multiplication

- Consider a number is an array:
`int[] B = { 9, 0, 1 };`
- Base10 Algorithm:
`sum = 0;`
`for (d = 0 ; d < 3 ; d ++) {`
 `sum += a * B[d];`
 `a = a * 10 ; // Base 10 shift left`
`}`
- Complexity: $O(\text{\#digits})$
 - For 2^{32} , at most 10 iterations

reframe:

	013	(a)	
*	109	(b)	

	117	(a*1)	* 9
+	000	(a*10)	* 0

	117		
	01300	(a*100)	* 1

	1417		

original:

	013	(a)
*	109	(b)

	117	(a*9)*1
+	000	(a*0)*10

	117	
	01300	(a*1)*100

	1417	

Note: commuta

Algorithm for Binary Multiplication

- Base 2 Algorithm:

```

sum = 0;
for (d = 0 ; d < 3 ; d ++ ) {
    if (B[d] == 1) {
        sum += a * B[d];
    }
    a = a * 2; // Base 2 shift left
    a = a << 1;
}
    
```

- Complexity: $O(\text{word_size})$

- For MIPS, at most 32 iterations

original:

	0010	(a = 2)
*	1011	(b = 11)

	0000 0010	(a*2 ⁰)* 1
+	0 0100	(a*2 ¹)* 1

	0000 0110	
+	00 0000	(a*2 ²)* 0

	0000 0110	
+	001 0000	(a*2 ⁴)* 1

	0001 0110	(a*b = 10)

	0010	(a = 2)
*	1011	(b = 11)

	0000 0010	(a*1)* 2 ⁰
+	0 0100	(a*1)* 2 ¹

	0000 0110	
+	00 0000	(a*0)* 2 ²

	0000 0110	
+	001 0000	(a*1)* 2 ⁴

	0001 0110	(a*b = 22)

Algorithm for Binary Multiplication

- Use the register as an stack
- Base 2 Algorithm:

```
sum = 0;
for (; b != 0; ) {
    bit = pop(b);
    if (bit == 1) {
        sum += a;
    }
    a = a << 1;
}
```
- Complexity: $O(\text{word_size})$

reframe:

	0010	(a = 2)
*	1011	(b = 11)

	0000 0010	(a*2 ⁰)* 1
+	0 0100	(a*2 ¹)* 1

	0000 0110	
+	00 0000	(a*2 ⁴)* 0

	0000 0110	
+	001 0000	(a*2 ⁸)* 1

	0001 0110	(a*b = 10)

original:

	0010	(a = 2)
*	1011	(b = 11)

	0000 0010	(a*1)* 2 ⁰
+	0 0100	(a*1)* 2 ¹

	0000 0110	
+	00 0000	(a*0)* 2 ⁴

	0000 0110	
+	001 0000	(a*1)* 2 ⁸

	0001 0110	(a*b = 22)