

Goal: A stronger understanding of signed binary multiplication.

Objective: To implement a C program that performs signed binary multiplication.

Assignment: (Read the extra credit part before beginning!)

Note: Be sure to #include <stdint.h> so that you can use the new C99 data types for integers.

Write a C program that does the following:

1. Read an unsigned decimal integer into a variable of type `int` using `scanf` with the `%d` format specifier. Use a cast to copy this into a variable of type `int8_t`. This will be your eight-bit multiplier.
2. Repeat step 1 to get an eight-bit multiplicand.
3. Compute the product of the two operands without using the multiply operator (`*`). No loop shall repeat for more than eight iterations. The product should be of type `int16_t`.

Note: Unlike the previous labs, this lab does NOT use arrays. Instead, your code should operate directly on the bits within the eight and 16-bit operands. For example, when you need to shift the bits to the left, use the left-shift (`<<`) operator instead of moving the contents of an array.

Hint: Read the document, "Converting an Unsigned Integer Product to a Signed Integer Product". First compute the 16-bit unsigned product, then modify it as described in the document to get the 16-bit signed product.

4. Display the resulting unsigned decimal product.

When you run your program, the screen should look something like:

```
Enter the multiplier: -23
Enter the multiplicand: 4
The corresponding product is: -92
```

Verify that your program operates correctly for several combinations of input values in the range -128 to +127, including end-of-range values. Demonstrate your program to the teaching assistant.

5% Extra Credit:

Organize your program into two functions:

1. A new function with two parameters – the multiplier and the multiplicand. Both should be of type `int8_t`. The product returned by the function should be of type `int16_t`.

Note: This function should NOT call `printf` or `scanf`.

Function prototype: `int16_t Multiply(int8_t multiplier, int8_t multiplicand);`

2. The main program, which should call the other function to compute the result as displayed in the sample output shown above.