Unsigned 8-bit Multiply with 8-bit result in 16-bit

Given variables A and B, each holding an 8-bit signed 2's complement number. Write a program to find the maximum value and put into variable C. Example if A > B then C = A.
C = A x B
Simulation of the unsigned problem C = 25 x 50, where the answer should equal 1250 (0x04E2).

```
reset:
    ;Initialize SRM Variables
    clr   r16
    sts   A, r16
    sts   B, r16
    sts   C, r16
    sts   C+1, r16
loop:
    ; Test Max1
    ldi   r16, 0x19
    sts   A, r16
    ldi   r16, 0x32
    sts   B, r16
    rcall Mul8x8_16
    rjmp loop
```

Watch

Name	Value	Type	Location
A	0x19 '┤'	SRAM Location	0x0100 [SR
B	0x00 ''	SRAM Location	0x0101 [SR
C	0x00 ''	SRAM Location	0x0102 [SR

Memory

Data | 8/16 | abc. | Address: 0x100

```
000100 19 00 00 00 00 00 00 00 00 00  ..........
00010A 00 00 00 00 00 00 00 00 00 00  ..........
000114 00 00 00 00 00 00 00 00 00 00  ..........
00011E 00 00 00 00 00 00 00 00 00 00  ..........
000128 00 00 00 00 00 00 00 00 00 00  ..........
```

Figure 1: Start of program with variable A initialized to 0x19 (25_{10})

```
reset:
    ;Initialize SRM Variables
    clr   r16
    sts   A, r16
    sts   B, r16
    sts   C, r16
    sts   C+1, r16
loop:
    ; Test Max1
    ldi   r16, 0x19
    sts   A, r16
    ldi   r16, 0x32
    sts   B, r16
    rcall Mul8x8_16
    rjmp loop
```

Watch

Name	Value	Type	Location
A	0x19 '┤'	SRAM Location	0x0100 [SR
B	0x32 '2'	SRAM Location	0x0101 [SR
C	0x00 ''	SRAM Location	0x0102 [SR

Memory

Data | 8/16 | abc. | Address: 0x100

```
000100 19 32 00 00 00 00 00 00 00 00  .2........
00010A 00 00 00 00 00 00 00 00 00 00  ..........
000114 00 00 00 00 00 00 00 00 00 00  ..........
00011E 00 00 00 00 00 00 00 00 00 00  ..........
000128 00 00 00 00 00 00 00 00 00 00  ..........
```

Figure 2: variable B is initialized to 0x32 (50_{10})

```
reset:
    ;Initialize SRM Variables
    clr   r16
    sts   A, r16
    sts   B, r16
    sts   C, r16
    sts   C+1, r16
loop:
    ; Test Max1
    ldi   r16, 0x19
    sts   A, r16
    ldi   r16, 0x32
    sts   B, r16
    rcall Mul8x8_16
    rjmp loop
```

Watch

Name	Value	Type	Location
A	0x19 '┤'	SRAM Location	0x0100 [SR
B	0x32 '2'	SRAM Location	0x0101 [SR
C	0xE2 'â'	SRAM Location	0x0102 [SR

Memory

Data | 8/16 | abc. | Address: 0x100

```
000100 19 32 E2 04 00 00 00 00 00 00  .2â.......
00010A 00 00 00 00 00 00 00 00 00 00  ..........
000114 00 00 00 00 00 00 00 00 00 00  ..........
00011E 00 00 00 00 00 00 00 00 00 00  ..........
000128 00 00 00 00 00 00 00 00 00 00  ..........
```

Figure 3: End of program with variable C containing 0x04E2 (32_{10}). Byte ordering is little endian.