Conversion (Qm.n)

INT → FX

```
y = x << n;
```

FX → INT

```
k = 1 << (n-1);
y = (x + k) >> n;
```

FP → FX

```
#include <math.h>
y = round( x * (1 << n) );
```

FX → FP

```
y = (double)x / (1 << n);
```

Le mete el 1<<(n-1) para aproximar los .5 al entero siguiente

Rounding (Qm.n)

Truncation

```
int floor(int x) {
    return x & ~((1<<n)-1);
}
```

1111100000 quedan n ceros al final que truncan x.

Ceiling

```
int ceil(int x) {
    return floor(x + ((1<<n)-1));
}
```

Rounding

```
int round(int x) {
    return floor(x + (1<<(n-1)));
}
```

Operations (Qm.n)

Zero

```
z = 0;
```

Unit

```
z = 1 << n;
```

One half

```
z = 1 << (n-1);
```

Negation

```
z = -x;
```

Absolute value

```
z = abs(x);
```

Operations (Qm.n)

Addition

```
z = x + y;
```

Subtraction

```
z = x - y;
```

Multiplication by an integer

```
z = x * i;
```

Multiplication

```
k = 1 << (n-1);
z = ( x * y + k ) >> n;
```

Division

```
z = (x << n) / y;
```