

Decimal No to Decimal Signed 2's Complements

Hexadecimal to Decimal converter

From

To

Hexadecimal

Decimal

Enter hex number

FE8D

16

= Convert

✕ Reset

↕ Swap

Decimal number (5 digits)

65165

10

Decimal from signed 2's complement (3 digits)

-371

10

Binary number (16 digits)

1111111010001101

2

Decimal calculation steps

$$(FE8D)_{16} = (15 \times 16^3) + (14 \times 16^2) + (8 \times 16^1) + (13 \times 16^0) = (65165)_{10}$$

Ref: <https://www.rapidtables.com/convert/number/hex-to-decimal.html> (<https://www.rapidtables.com/convert/number/hex-to-decimal.html>)

Now objective to covert from 65165 to -371

two approaches to the problem:

- The first assuming you are doing two's complement arithmetic all the time, in which case the digit adding must be done with sign.
- The second assuming you only parse unsigned values and retaining the sign to make the sign exchange at the end.

Probably both approaches will lead to almost the same efficiency and be compiled into very similar code. I have no preference for any of them.

```
int decode(char *str, int base)
{
    int result = 0,
    c,
    neg = FALSE;

    /* skip whitespace, delete this if you don't
    * want to cope with whitespace */
    for (; isspace(c = *str); str++) {
        continue;
    }

    if (*str == '-') {
        neg = TRUE; /* negative */
        str++; /* skip it */
    }

    /* the next characters might be all digits */
    for (; isdigit(c = *str); str++) {

        /* multiply by the base */
        result *= base;

        /* add positive for positives and
        * subtract it for negatives */
        int d = c - '0'; /* convert c to the digit value */

        /* negative if number is negative */
        if (neg) d = -d;

        /* and add/subtract it */
        result = result + d;
    }

    /* :) got it!! */
    return result;
}
```

```

int decode(char *str, int base)
{
    int result = 0,
    c,
    neg = FALSE;

    /* skip whitespace, delete this if you don't
    * want to cope with whitespace */
    for (; isspace(c = *str); str++) {
        continue;
    }

    if (*str == '-') {
        neg = TRUE; /* negative */
        str++; /* skip it */
    }

    /* the next characters might be all digits */
    for (; isdigit(c = *str); str++) {

        /* multiply by the base */
        result *= base;

        /* add positive for positives and
        * subtract it for negatives */
        int d = c - '0'; /* convert c to the digit value */

        /* and add/subtract it */
        result = result + d;
    }

    /* :) got it!! */
    return neg ? -result : result;
}

/* Test code in arduino */

#include <stdio.h>
#include <ctype.h>
#include <stdlib.h>

void setup() {
    // put your setup code here, to run once:

}

```

```
void loop() {  
  // put your main code here, to run repeatedly:  
  Serial.begin(2000000);  
  Serial.print("\n 65165 Int to 2s complement :");  
  char test[] = "65165";  
  Serial.println(decode(test, 10));  
  delay(1000);  
  
}
```

```
int decode(char *str, int base)  
{  
  int result = 0;  
  char c;  
  bool neg = 0;  
  
  /* skip whitespace, delete this if you don't  
  * want to cope with whitespace */  
  for (; isspace(c = *str); str++) {  
    continue;  
  }  
  
  if (*str == '-') {  
    neg = 1; /* negative */  
    str++; /* skip it */  
  }  
  
  /* the next characters might be all digits */  
  for (; isdigit(c = *str); str++) {  
  
    /* multiply by the base */  
    result *= base;  
  
    /* add positive for positives and  
    * subtract it for negatives */  
    int d = c - '0'; /* convert c to the digit value */  
  
    /* and add/subtract it */  
    result = result + d;  
  }  
  
  /* :) got it!! */  
  return neg ? -result : result;  
}
```