

## 1 データ型のサイズを調べる

「C 言語」の 21 頁に左に int 型の変数についての説明があります。  
色々な開発環境があるので、教科書通りではありません。  
そこで、sizeof() 関数を使って各変数のサイズを調べてみます。

### 1.1 データ型のサイズを調べる varsize.c

```
/*
   データ型のサイズを調べる。
   varsize.c
*/

#include <stdio.h>

int main()
{
    printf("char 型      %d バイト\n", sizeof(char));

    printf("short 型     %d バイト\n", sizeof(short));

    printf("int 型       %d バイト\n", sizeof(int));

    printf("long 型      %d バイト\n", sizeof(long));

    printf("ポインタ    %d バイト\n", sizeof(void));

    printf("long long 型 %d バイト\n", sizeof(long long));

    return 0;
}
```

### 1.2 結果

```
char 型      1 バイト
short 型     2 バイト
int 型       4 バイト
```

long 型	4 バイト
ポインタ	1 バイト
long long 型	8 バイト

int 型と long 型のサイズが、同じ 4 バイトになっています。

### 1.3 long long 型

C 言語が初めて作成されてから、現在までかなりの時間が経過しています。それに従い計算機も進化してきました。例えば、64 bit のアドレス空間を持つ CPU の登場です。64bit あれば 18E(エクサ) もの巨大な数を表すことができます。従来の C 言語のコンパイラではそのような巨大な数を表現することは想定していませんでした。そこで、今回新たに long long int という、long 型よりもさらに大きな数を表現できる型が新設されました。