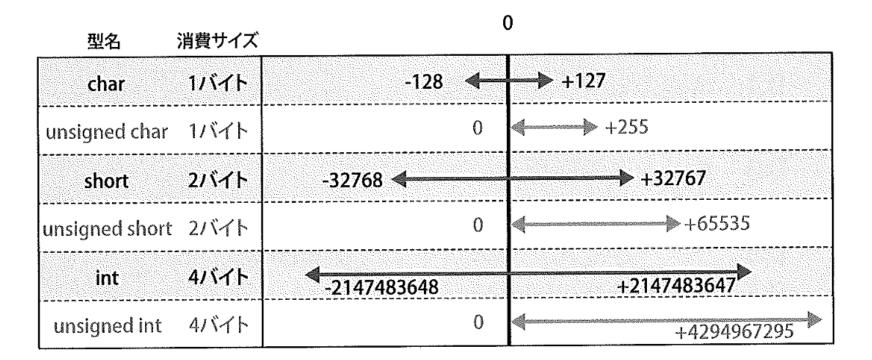
情報システムプログラミング**I** (**16**回目)

2024年10月2日(水)3~4限

授業内容

- 講義内容(教科書の690~694ページ)
 - ▶情報のビット表現と型
 - ▶ビット演算
- 演習課題

- ■符号あり型と符号なし型
 - 負の値が不要な場合はunsignedを付けた型を利用する
 - ▶ 負の値を扱えない代わりに上限値を上げることが可能



- ■ビットの解釈の違い
 - 符号なし型では単純に右端から 左端に向かってビットを使う
 - 符号あり型では先頭の1ビットを 符号ビットとして使う
 - ▶ 0だと正の値, 1だと負の値



D). [E]	char 型の値			
ビット列	unsigned	signed		
00000000	0	0		
00000001	1	1		
00000010	2	2		
00000011	3	3		
:	:	:		
01111111	127	127		
10000000	128	-128		
10000001	129	-127		
:	:	*		
11111101	253	-3		
11111110	254	-2		
11111111	255	! [

- ■バイトオーダー
 - 2バイト以上のデータにおけるバイト単位のデータの並び順➤ エンディアンともいう
 - 右側から並べる方法をビッグエンディアン,左側から並べる 方法をリトルエンディアンという
 - ▶ 例えば、256 (10進数) /0000000 10000000 (2進数) に おける各エンディアンは以下の通り
 - ✓ ビッグエンディアン:0000001 0000000
 - ✓ リトルエンディアン:00000000 00000001

- ■サイズが明確な整数型
 - ・以下のサイズ固定の整数型がC99より追加

サイズ	符号	型名	備考
8ビット(1バイト)	あり	int8_t	本書での char に相当
	なし	uint8_t	本書での unsigned char に相当
16 ビット (2 バイト)	あり	int16_t	本書での short に相当
	なし	uint16_t	本書での unsigned short に相当
32 ビット (4 バイト)	あり	int32_t	本書での int に相当
	なし	uint32_t	本書での unsigned int に相当
64 ビット (8 バイト)	あり	int64_t	本書での long に相当
	なし	uint64_t	本書での unsigned long に相当

• 【余談】C言語の最新規格はC17(C99の後継のC11の更に 後継)で、2024年中にはC17の後継としてC23が定まる予定

ビット演算

- ■ビット単位で情報操作を行う演算子
 - ビット単位の論理演算を行うビット論理演算子と、ビットの 並びを右または左に動かす(シフトする)演算を行うシフト 演算子がある

演算子	意味	使用例	解説
~	ビットごとの NOT	~a	a の各ビットの 0 と 1 を反転
&	ビットごとの AND	a&b	a と b のビット単位の AND ** ^{1、** 4}
	ビットごとの OR	a b	a と b のビット単位の OR ** ² ** ⁴
^	ビットごとの XOR	a ^ b	a と b のビット単位の XOR ** 3、 ** 4
<<	左シフト	a << b	aをbビット分、左へずらす
>>	右シフト	a >> b	aをbビット分、右へずらす

- ※1 ANDは、aとbが1なら結果は1、それ以外は0を返す。
- ※2 ORは、aかbが1なら結果は1、それ以外は0を返す。
- ※3 XORは、aとbのビットが異なれば1、等しければ0を返す。
- ※ 4 演算と同時に代入も行う &=、|=、 ^= 演算子も利用可能。

ビット演算

- ■「ビット単位の論理演算」と単なる「論理演算」の違い
 - 論理演算ではビット単位か否かで処理が変わるので注意!
 - ▶ビット単位の論理演算の例

```
4 \& 6 \Rightarrow 4 \qquad 4 \mid 6 \Rightarrow 6
(100 & 110 \Rightarrow 100) \quad (100 \left| 110 \Rightarrow 110)
```

▶論理演算の例

```
4 && 6 ⇒ 1 4 || 6 ⇒ 1
(真(非0) && 真(非0) ⇒ 真) (真 || 真 ⇒ 真)
```