



第2話 (ゾーンとパック)



ねえ、ねえ、キツネ！
 オイラがキーボードから **A** **1** とい
 うキーを押したら、その後どうなるの？



タヌキが **1** を押すと [0011 0001] というコードが記
 憶装置に運ばれ、 **A** を押すと [0100 0001] というコ
 ードが記憶装置に運ばれます。[0011 0001] の [0011]
 はゾーン部といい、数字を文字扱いにする為の方式です。
 ゾーン部 [0011] を外し、[0001] にすることをパックと
 いい、押された **1** を数値として計算に使えます。



[0011 0001] とか [0100 0001] は何の
 おまじないだ！



これは、2進数といい、コンピュータはON (1) とOFF
 (0) の回路なので全て2進数で表現されるのだ。[0011
 0001] は10進数で表現すると49、16進数ならば31
 です。2進数や16進数は自分で勉強してね。オイラはア
 ウトラインを教えるだけだから、自分でも勉強して。
 勉強とはシイルことだから、楽ばかりじゃないよ！



ついでに、数値扱いになっているデータをモニタ
 ーに出力する時には、ゾーン部 [0011] を付けな
 ければ表示されないことを付け加えておくね。
 パックにゾーン部を付けることをアンパックとも
 言うよ。



文字はキーに割り振られたコードのまま記憶されることはわかったけれど、数値はパック方式で記憶されるということで良いのかな。

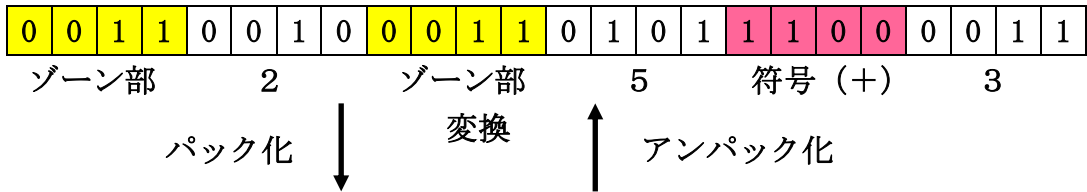


文字の記憶のされかたはタヌキの言う通りだが、数値はちょっと面倒なのだ。数値にはプラスとマイナスがあり、さらに整数もあれば少数点の付く実数というものがあるでしょう。記憶装置への数値の記憶のされかたは、大切なのでこれからコメントします。

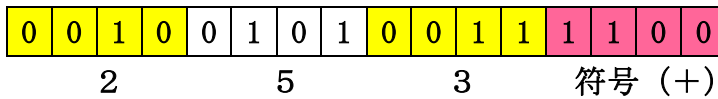
記憶装置での数値の記憶のされかた

正の整数（例：+253 の場合）

ゾーン 10 進数(文字扱い)

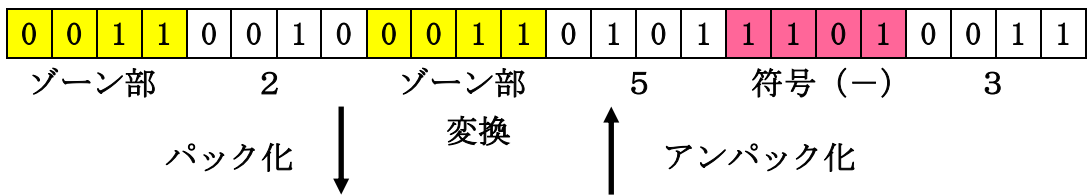


パック 10 進数(数値扱い)

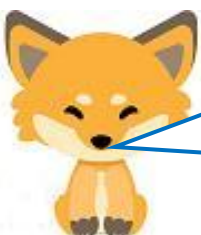
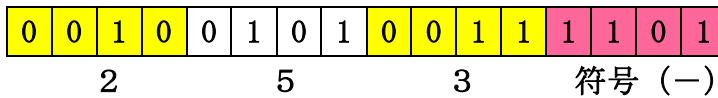


負の整数（例：-253 の場合）

ゾーン 10 進数(文字扱い)



パック 10 進数(数値扱い)

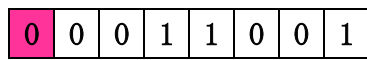


キーボードから入力された数字は必要に応じてゾーン 10 進数やパック 10 進数で記憶されますが、プログラムの指示に従って整数型、実数型（固定小数点、浮動小数点）で記憶される場合もあるよ。次にそれを説明するね。

プログラムの指示 (型) による記憶装置での数値の記憶のされかた

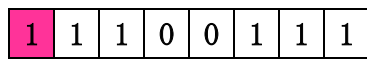
整数 (int) 型

正の整数 (例: +25) の記憶のされ方 (記憶容量 8 ビット = 1 バイトの場合)



↑
符号ビット (+)

負の整数 (例: -25) の記憶のされ方 (2 の補数)



↑
符号ビット (-)



チョイ待ち!

-25 を記憶させるのに、2 の補数で、なんだよ。説明の省略しすぎだろ。



面倒だから、補数については後で説明しようと思っていたのだが、2 進数でなく 10 進数で補数を簡単に説明するよ。負の数を補数にすると引き算を加算にすることができるのさ。演算装置に加算回路と減算回路があるが、加算回路の方が簡単な回路なので、負荷がかからず、処理も早くなるのだ。

では、 $25 + (-25) = 0$ を補数計算で行うと $25 + 75 = 00$ となることで補数のイメージをつかんでくれ。

10 進数 (-25) の補数計算

$$\begin{array}{r} 99 \text{ (2桁の最大数)} \\ -25 \text{ (10進の正の数)} \\ \hline 74 \text{ (10進数の1の補数)} \\ + 1 \\ \hline 75 \text{ (10進数の2の補数)} \end{array}$$

$$25 + 75 = \cancel{1}00$$

2 の補数とは、1 の補数に 1 を加えること。2 桁の最大数に事前に 1 を加え、3 桁にすると 2 の補数になる。

$$\begin{array}{r} 100 \text{ (99に先に1プラスした3桁数)} \\ - 25 \\ \hline 75 \text{ (10進数の2の補数)} \end{array}$$

2 桁の計算なので桁上がりの 1 を削除すると 00 (つまり 0) になる。



(25 - 25) は0だよ。10進数の2の補数を用いた計算 (25 + 75) は桁上がりの3桁目は記憶できない、と考えれば結果は0になるよ。これが減算を加算で処理できるということなんだ。ちょっと誤魔化したけどタヌキ気がついた？じゃ、2進数の2の補数で同じことをやってみよう。

2進数の2の補数計算

00011001 (10進数の+25)

11111111 (8桁の最大数)
 -00011001 (2進の正の数)

11100110 (2進数の1の補数)
 + 1
 11100111 (2進数の2の補数)

00011001 (+25)
 +11100111 (-25)

~~10000000 (0)~~

8ビットの記憶領域なので桁上がりの9ビット目は桁落ちします。



2進数の1の補数 (11100110) を求めるのに、減算回路を使っているよね。これじゃ加算回路だけで処理することができないのでは。オイラを化かしているな！



タヌキ、鋭いね！
 コンピュータでは、2進数の1の補数 (11100110) は減算でなく (00011001) を反転させることによって求めているんだ。その後に1をプラスするので、1の補数を経て2の補数を求めるという順序を変えることはできないのさ。
 ところで、10進数の説明には無理があるんだ。補数計算後の100は、補数で求められるので、補数を元に戻すので100 - 100 = 0とするのが正解なのだ。



オイラ、ハワイに行った時にお土産にナッツチョコを2ドルで買って、10ドル札を渡したんだ。その時、売り場のお姉さんが、オイラに向かって手を開いてください、というので開いたら、その手のひらに3ドル、4、5、6、7、8、9、10ドルと1ドル札を8枚のせてくれたんだ。この時、オイラはこのお姉さん、引き算を加算に代えてしまった、と感心したんだ。



キツネよ！
葉っぱの10ドル札を渡して、2ドルのナッツチョコと8ドル札をせしめたわけじゃないだろな。



タヌキ、深く考えない。
次は、演算装置での加算を考えよう。
その前に、大切なことが1つあります。
記憶装置にはプログラムを記憶する部分とプログラムが使用するデータの部分が分かれて存在することを覚えておいて。じゃあ

第3話

記憶装置で

プログラムを記憶する領域

記憶装置で

データ（使用する）を記憶する領域