5章 コンピュータのしくみ

26　2進法による数の表現と計算

基本問題・解答

1
(1) 2進法 1101　16進法 D
(2) 2進法 11011　16進法 1B
(3) 2進法 1111110　16進法 7E
(4) 2進法 1000011100　16進法 21C

解説　10進法の数を2進法の数や、16進法の数に変換するには、2進法の場合は2で、16進法の場合は16で割っていくことで余り求める。16で割ったときの余りについて、10進法の10～15は16進法のA～Fとすることに注意。
(1) 13(10) = 1101(2) = D(16)
(2) 27(10) = 11011(2) = 1B(16)
(3) 126(10) = 1111110(2) = 7E(16)
(4) 540(10) = 1000011100(2) = 21C(16)

2
(1) 11111110　(2) 100100101
(3) 1011000　(4) 11110

解説　桁上がり、桁下がりが生じる計算に注意。とくに2進法の数での減算では、桁借りをするときには10 − 1 = 1、10 − 1 − 1の2パターンがあることに気をつける。
(1) 10011001 + 01100101 = 11111110
(2) 11011000 + 01001101 = 100100101
(3) 10110101 − 01011101 = 1011000
(4) 11011000 − 10111010 = 11110

3
(1) 1011　(2) 10011
(3) 10100011　(4) 01000110

解説　2の補数は、①各ビット列を反転する、②1を足す。
(1) 0101 → 1010（反転）→ 1011（1を足す）
(2) 1101 → 01101（5ビット化）→ 10010（反転）→ 10011（1を足す）
(3) 01011101 → 10100010（反転）→ 10100011（1を足す）
(4) 10111010 → 01000101（反転）→ 01000110（1を足す）

4
(1) C5　(2) 90

解説　2の補数を求める場合の①反転、②1を足すと同じ考え方で求められる。
(1) 3B → C4（反転）→ C5（1を足す）
(2) 70 → 8F（反転）→ 90（1を足す）

5
(1) 0.01　(2) 10.11　(3) 1011.1011

解説　10進法の数 $0.1_{(10)}$ は2進法では 1×2^{-1}、$0.01_{(10)}$ は 1×2^{-2}、…と考える。
(1) 0.25は 1 × 2 − 2 → 0.01
(2) 2.75は整数部を2、小数部を0.50 + 0.25、つまり $1 \times 2^{-2} + 1 \times 2^{-2}$ と考える。
(3) 11.6875は整数部を11、小数部を0.5 + 0.125 + 0.625、つまり $1 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4}$ と考える。

実践問題・解答

1
(1) 4E　(2) 1100001

解説　(1) 2進数4桁が16進数1桁に相当するので、2進数の下位から4桁ずつ16進数に変換する。100と1110にわけ、それぞれを10進数にすると4と14になる。これを16進数にすると4とEになる。
(2) 1001110より3つ大きいので2進数11を足せばよい。

2
ア 31　イ 16　ウ 15　エ 11010　オ 4

解説　5ビットで計算することが前提となっている。0と正の整数を表現する場合、$11111_{(2)}$ が最大数になり、これを10進法に変換すれば15(ア)となる。負の数は2の補数で表現する。負の数は $11111_{(2)}$ が $-1_{(10)}$ に相当し、$10000_{(2)}$ が負の数のうち絶対値が最も大きくなる。2の補数の求める操作を逆に行う。$10000_{(2)}$ から1を引いた $01111_{(2)}$ のビットを反転すると $10000_{(2)}$ となる。これは $16_{(10)}$ に相当するので $10000_{(2)}$ は $-16_{(10)}$(イ)となる。正の数の最大値は $01111_{(2)}$ なので15(10)(ウ)である。$-6_{(10)}$ の絶対値6を2進数 $00110_{(2)}$ で表す。次に、ビット反転し $11001_{(2)}$、さらに1を足すと $11010_{(2)}$(エ)を得ることができる。$01100_{(2)}$ と $11000_{(2)}$ を足すと $100100_{(2)}$ となるが、5ビットで計算することが前提となるため6桁目は無視した $00100_{(2)}$(オ)が計算結果となる。この計算は、10進法で考えると12-8=4を計算していること相当する。

3
(1) 符号部 1、指数部 10000001、仮数部 11101100
(2) 符号部 0、指数部 10000010、仮数部 01000110

解説　符号部は負の数なら1、正の数なら0なので、(1)の符号部は1、(2)の符号部は0である。符号を除いた数を2進数で表すと(1)は111.1011、(2)は $1010.\dot{0}01\dot{1}$（傍点は循環小数を表す）。これらを浮動小数点数で表すと、(1)は 1.111011×2^2、(2)は $1.010\dot{0}01\dot{1} \times 2^3$ となる。よって、指数部は(1)が2+127=129、(2)は3+127=130をそれぞれ2進法に変換したものになる。仮数部は浮動小数点数の小数部を8ビットまで表したものが答えとなる(8桁以下なら0を付け加え8桁にする)。

4
(1) $0.0\dot{1}00\dot{1}$
(2) 00011001
(3) -0.00234375
(4) ①

解説　(1) 10進法の小数を2進法に変換するには2を掛けて整数部を記録する。
0.3 × 2 = 0.6　2進法 0.0
0.6 × 2 = 1.2　2進法 0.01
0.2 × 2 = 0.4　2進法 0.010
0.4 × 2 = 0.8　・2進法 0.0100
0.8 × 2 = 1.6　2進法 0.01001
0.6 × 2 = 1.2　2進法 0.010011 以降同じパターンが続く
(2) 小数0.3と0.2を2進法の小数第8位まで求め、これを引き算する。

10進法		2進数	
	0.3		0.01001100
	−0.2		− 0.00110011
	0.1		0.00011001

(3) (2)で求めた2進数の減算の結果を10進法に変換し、10進数の減算結果0.1との差を計算する。
(4) 計算機が扱える桁数に限りがあるため、循環小数のように無限に続く小数を計算する際に丸め誤差が生じる。

27　総和、最大･最小値のアルゴリズム

基本問題・解答

1
(1) 13
(2) ア nokori　イ t　ウ a * Kinsyu[t]
エ a　オ 1

解説　(1) 36723円を支払うときの紙幣と硬貨の数は、残り金額を金種ごとの支払い金額を引いた残金を次に大きな金額の紙幣または硬貨で計算することを繰り返す。

36723−	10000×3=	6723	1万円札3枚使用
6726−	5000×1=	1723	5千円札1枚使用
1723−	1000×1=	723	1千円札1枚使用
723−	500×1=	223	500円硬貨1枚使用
223−	100×2=	23	100円硬貨2枚使用
23−	10×2=	3	10円硬貨2枚使用
3−	1×3=	0	1円硬貨3枚使用

合計13枚

(2) n円に対して、金額の大きい金種の紙幣または硬貨から順に(1)で行ったように使用枚数を考え、残り金額を求める計算を反復する。変数nokori,s,t,aの役割を考える。nokoriは残金であり、6行目でn円を初期値として代入している。sは16行目から使用した紙幣と硬貨の数であることがわかる。配列Kinsyuのうち計算する金種を指定するためには添字の値を0から1ずつ増やす必要があり、これを変数tを使って実現している。
9行目の反復条件は残り金額が0円になるまで計算することになるので、nokori > 0が反復条件である。10行目はKinsyu[t]を使う枚数を計算しaに代入している。12行目は新たな残り金額の計算である。13行目は使用枚数をa増やし、14行目は次の金種を指定するためtの値を1増やす。

▶ 再帰関数
再帰関数とはプログラミングの手法の1つで、プログラムの中に自分自身の呼び出しが含まれているものをいう。再帰関数は、繰り返し関数と同様に、同様な処理を複数回行う場合に利用される。

2
(1) Tensu[29][2]
(2) ア kyokasu − 1　イ Tensu[j][i]
ウ t > -1 または t >= 0　エ t
オ saitei　カ saiko　キ gokei / juken

解説　(1) 添字は0から始まるので30番目の生徒の添字は29、国語は3番目なので添字は2となる。