問題２　次の数値表現に関する記述を読み，各設問に答えよ。

コンピュータで扱う数値には，小数点以下の値を持たない整数型や小数点以下を扱える実数型がある。整数型を扱う場合に使用するのが固定小数点数であり，実数型を扱う場合に使用するのが浮動小数点数である。

＜設問１＞　次の固定小数点数に関する記述中の[　　　　]に入れるべき適切な値を解答群から選べ。

固定小数点数とは，小数点を決められた場所に固定して表現するものである。整数型として扱う場合，最右端ビットの右側に小数点位置がある。

0	0	1	0	…	0	1	1	0	1

▲小数点位置

図１　固定小数点数

例えば，８ビットの固定小数点数で正数のみを扱うとすれば，その最小値は２進数で 00000000 であり，最大値は 11111111 である。それぞれ 10 進数では０と[(1)]である。

負数を扱う場合は，先頭ビットを符号ビットとした２の補数表現を使う。

［２の補数を求める手順］

① 絶対値を２進数に変換する。

② 各ビットの０と１を反対にする(１の補数)。

③ １を加える(２の補数)。

例えば，10 進数の-28 を８ビットの２進数で表現する場合，絶対値の+28 は 00011100 であり，２の補数で表現された-28 は，[(2)]である。

このように負数を２の補数で表現する場合，８ビットで扱える最小値は２進数で[(3)]であり，最大値は[(4)]である。

(1) の解答群

ア．127　　イ．128　　ウ．255　　エ．256

(2) の解答群

ア．11100000　　イ．11100011　　ウ．11100100　　エ．11100111

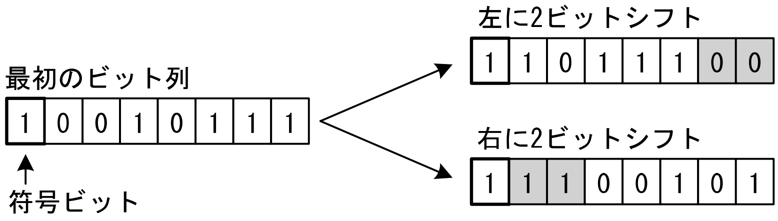
(3)，(4) の解答群

ア．00000000　　イ．01111111　　ウ．10000000　　エ．11111111

＜設問２＞　次の算術シフトに関する記述中の［　　　　］に入れるべき適切な値を解答群から選べ。

算術シフトは符号ビットを除いてシフト（桁移動）され，左にシフトした場合に空いた右側のビットには 0 が，右にシフトした場合に空いた左側のビットには符号ビットと同じビットが格納される。

なお，算術シフトは，n ビット左にシフトすると 2^n 倍，n ビット右にシフトすると $1/2^n$ 倍になる。例えば，2 ビット左にシフトすると 4 倍，2 ビット右にシフトすると 1/4 倍になる。



注）網掛け部分がシフトにより空いたビット位置

図２　算術シフトの例

ここで，2 進数で 11110000(10 進数で−16)を，左に 2 ビットシフトすると［　(5)　］(10 進数で−64)であり，右に 2 ビットシフトすると［　(6)　］(10 進数で−4)である。

(5)，(6) の解答群

ア．00110000　　イ．00111100　　ウ．11000000　　エ．11111100

＜設問３＞　次の浮動小数点数に関する記述中の［　　　　］に入れるべき適切な字句を解答群から選べ。

浮動小数点数とは，数値を $(-1)^{符号}$×仮数×基数^{指数} として表現するものである。ここでは 32 ビット（単精度）の IEEE754 形式で説明する。

符号部	指数部	仮数部
1 ビット	8 ビット	23 ビット

図３　IEEE754 単精度浮動小数点数形式

・符号部は仮数部の符号を表し，非負の場合は 0，負の場合は 1 とする。
・指数部は 2 を基数とし，実際の値に 127 を加えたバイアス値とする。
・仮数部は 10 進数で 1 以上 2 未満になるように調整することで，2 進数で表現すると「1.XXX…」となり，そこから 1 を引いた値とする。この操作を正規化と呼ぶ。

例えば，2 進数の 10111 は，1.0111×2^4と調整し，符号部は 0，指数部は 4+127=131（2 進数で 10000011），仮数部は 1.0111 から 1 を引いた 0.01110…0（小数部分は全部で 23 ビット）となる（図４）。

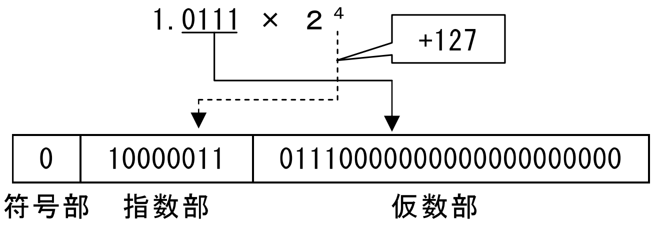


図４　IEEE754 形式で 2 進数の 10111 を表現した結果

同様に，10 進数の 0.625 を IEEE754 形式で表現すると，各部の値は 2 進数表現で次のようになる。

0	01111110	(7)

また，IEEE754 形式で次のように表現された浮動小数点数は，10 進数で表現すると (8) である。

0	10000001	10100000000000000000000

次に，A，B，C の三つの値に対する演算を考える。浮動小数点数は，演算する値の指数を大きい方に揃えて仮数の演算を行った後で正規化する。

A…1.010010000111×2^{24}

B…1.0100100001×2^{24}

C…1.1×2^{-1}

A－B の演算では，指数が同じであるから仮数をそのまま減算する。

1.010010000111－1.0100100001＝0.000000000011 となり，正規化すると 1.1×2^{13} である。このように仮数部の有効桁数が減少し精度が落ちることを (9) と呼ぶ。

A＋C の演算では，指数を大きい方に揃えるため，
C を(0.0000000000000000000000000011)×2^{24} に調整して加算するが，右端の 11 が表現できずに 0 を加算することになる。このように加算に C の値が反映されないことを (10) と呼ぶ。

(7) の解答群

ア．01000000000000000000000　　イ．01010000000000000000000

ウ．10100000000000000000000　　エ．11000000000000000000000

(8) の解答群

ア．2.5　　イ．5.0　　ウ．6.5　　エ．13.0

(9)，(10) の解答群

ア．切り上げ　　イ．切り捨て　　ウ．桁落ち　　エ．情報落ち