問8　符号付き２進整数の乗算（データ構造及びアルゴリズム）（H22秋·FE 午後問8）

【解答】

［設問1］　a－エ，b－オ，c－イ

［設問2］　d－ウ，e－オ，f－ウ，g－ア

【解説】

二つの整数M，Nを受け取り，その積M×Nの値を返すプログラムの問題である。積は，加減算とシフト演算を使って求める。問題では，二つのプログラムが与えられており，プログラム１は加算とシフト演算，プログラム２は加減算とシフト演算で積を求めている。プログラムでは，整数は２進数（負数は２の補数）で表現されており，また，算術シフトでビットをずらす処理もあり，システムの利用部門に所属する受験者はやや戸惑いがあったかもしれない。しかし，これらの内容はコンピュータの基礎理論として学習の範囲に入っているため，しっかりと理解してほしい内容である。

設問はプログラムのトレースとプログラムの処理内容に関する問題で構成されている。各プログラムがどのように処理を行っているのかを，問題文の意味を理解し，実行例と対応付けて考えることで，正解を導くことが可能である。アルゴリズム問題は，手間を惜しまず，根気良く手を動かして一つずつ確認していくことが大切である。

〔プログラム１〕
```
      ○プログラム１(M, N)
      ○符号付き２進整数型： M, N, R
      ○整数型： L
①→ ・ Mを５ビットの符号付き２進整数に拡張
②→ ・ Nを８ビットの符号付き２進整数に拡張
③→ ・ Rのビット番号７～０にNを複写
④→ ・ Rのビット番号12～８を０で初期化
    ■ L: 1, L ≦ 8, 1
⑤→ │ ▲ Rのビット番号０のビットが１
⑥→ │ │ ・ Rのビット番号12～８の内容にMの値を加算
    │ ▼
⑦→ │ ・Rの全13ビットを右に１ビット算術シフト   /* 空いたビット位置には */
    ■                                         /* 符号と同じものが入る */
⑧→ ・return( Rのビット番号７～０の内容 )        /* 返却値（括弧内）を返す */
```

〔プログラム２〕
```
      ○プログラム２(M, N)
      ○符号付き２進整数型： M, N, R
      ○整数型： L
①→ ・ Mを５ビットの符号付き２進整数に拡張
②→ ・ Rのビット番号４～１にNを複写
③→ ・ Rのビット番号９～５及び０を０で初期化
    ■ L: 1, L ≦ 4, 1
④→ │ ▲ Rのビット番号１のビットが０ かつ Rのビット番号０のビットが１
⑤→ │ │ ・ Rのビット番号９～５の内容にMの値を加算
    │ ▼
⑥→ │ ▲ Rのビット番号１のビットが１ かつ Rのビット番号０のビットが０
⑦→ │ │ ・ Rのビット番号９～５の内容からMの値を減算
    │ ▼
⑧→ │ ・Rの全10ビットを右に１ビット算術シフト   /* 空いたビット位置には */
    ■                                         /* 符号と同じものが入る */

⑨→ ・ return( Rのビット番号８～１の内容 )       /* 返却値（括弧内）を返す */
```

２進数は左や右にビットをシフトすることで2^n倍の演算が行える。例えば，２進数の11（10進数の３）を左に１ビットシフトすると110（10進数の６）となり，元の値の２倍になる。シフト演算だけでは2^n倍の演算しか行えないが，次に示すように2^n倍の演算と組み合わせてそれ以外の演算もできる。シフト演算には，論理シフトと算術シフトがあるが，補数を用いた負数の演算には算術シフトを用いる。算術シフトでは，符号ビット（左端のビット）を固定し，残ったビットを左右にシフトする。左シフトの場合は空いたビット位置には０を，右シフトの場合は空いたビット位置には符号と同じものを入れる。

＜10進数の５を３倍する場合（５ビットで表現）＞
```
    0 0 1 0 1   (5)10  ※(n)10は10進数の値を表す。
    0 1 0 1 0   左に１ビット算術シフトし，元の値を２倍する。(10)10

    0 1 0 1 0   ２倍した値に
 +  0 0 1 0 1   元の値を加算する。
 ─────────────
    0 1 1 1 1   元の値の３倍になる。(15)10
```

プログラムでは右に１ビット算術シフトして演算しているため，少し戸惑ったかもしれないが，次のように考えるとこれと同じ結果となる。

＜プログラムでの処理＞
```
    0 1 0 1 0   (5)10  ※最下位のビットは予備のビットである。
    0 0 1 0 1   右に１ビット算術シフトする。空いたビット位置には符
                号と同じ０が入る。
    0 0 1 0 1   右に１ビット算術シフトした値の左から４ビットに元の
 +  0 1 0 1 ↓   値を加算する。右端のビットはそのまま付加する。
 ─────────────
    0 1 1 1 1   (15)10のビット列ができ上がる。
```

右に１ビット算術シフトした値の左から４ビットに元の値を加算する部分を，次のように０を補って考えると，プログラムでは加算する方の値が2^n倍された値になっていることが分かる。

```
    0 0 1 0 1   右に１ビット算術シフトした値は(5)10になる。
 +  0 1 0 1 0   元の値に０を補うと加算する値は２倍の(10)10になる。
 ─────────────
    0 1 1 1 1   (15)10になる。
```

［設問1］

・空欄a，b：プログラムの実行例の○付き数字は，プログラムの○付き数字と対応しているので，図２の実行例とプログラム１の処理内容を対応付けてトレースする。処理④を実行後のR（0000000000011）のビット番号０は１なので，処理⑤の条件式が成り立つため，処理⑥でRのビット番号12～8の内容（00000）にMの値（11011）を加算し，Rは1101100000011となる。続いて，処理⑦でRの全ビットを右に１ビット算術シフトするので，Rは1110110000001となる。空欄aはRのビット番号12～7の内容であるため，「111011」の（エ）が正解となる。

繰返し処理は８回実行するため，再び処理⑤の条件式に戻る。Rのビット番号０は１なので，処理⑤の条件式が成り立ち，処理⑥でRのビット番号12～8の内容（11101）にMの値（11011）を加算し，Rは1100010000001となる（けたあふれは無視）。続いて，処理⑦でRの全ビットを右に１ビット算術するので，Rは1110001000000となる。空欄bはRのビット番号12～6の内容であるため，「1110001」の（オ）が正解となる。

・空欄c：図３の実行例とプログラム２の処理内容を対応付けてトレースする。処理③を実行後のR（0000000110）のビット番号１は１，ビット番号０は０なので，処理⑥の条件が成り立つため，処理⑦でRのビット番号9～5の内容（00000）からMの値（11011）を減算し，Rは0010100110となる。続いて，処理⑧でRの全ビットを右に１ビット算術シフトするので，Rは0001010011となる。繰返し処理は４回実行するため，繰返し処理の先頭の処理④の条件式に戻るが，Rのビット番号１と０はいずれも１なので，処理④，処理⑥の条件は成り立たない。よって，加減算は実行されずに処理⑧でRの全ビットが右に１ビット算術シフトされ，Rは0000101001となる。空欄cはRのビット番号9～3の内容であるため，「0000101」の（イ）が正解である。

［設問2］

・空欄d：各選択肢の内容を検証する。

ア：加算処理はRのビット番号０のビットが１の場合に実行される。プログラム１では，ビット番号０のビットが１かどうかの判断は８回実施されるが，Rの下位８ビットにはNの値が格納されており，加算処理の回数はMの値ではなく，Nの値によって決まる。

イ：Nの値が１のときは，Rのビット番号０のビットが一度だけ１になるため，ループ内の加算処理の実行回数は１となる。

ウ：プログラム１では，繰返し処理の中でシフト演算は１回しか実行されていない。よって，中間結果のシフトとNのビットの順次取出しは，１回のシフトで済ませている。

エ：プログラム１で取り出すビットは右端のビットであり，Nの値の符号検査は行っていない。

したがって，（ウ）が正解である。

・空欄e：プログラムが受け取るNは４ビットである。特に効果があるのはN<0のときなので，Nの範囲は－8≦N≦－1となる。プログラム１では処理を行う際にNを８ビットに拡張するため，－8と－1を２の補数で表すと11111000，11111111となる。加算処理はRのビット番号０のビットが１のときに行うため，１であるビットが最も少ない－8の場合が最も加算処理が少なく，実行回数は５回になる。したがって，空欄eは（オ）が正解である。

・空欄f：プログラム２では，Rのビット番号１のビットが０かつビット番号０のビットが１の場合に加算，Rのビット番号１のビットが１かつビット番号０のビットが０の場合に減算を行い，ビット番号１とビット番号０の値が同じ場合には加減算は行わない。つまり，ビットの値が０と１とが交互に並んでいる場合に加減算処理が最も多くなる。N<0の場合を考えるので，符号ビットは１になるため，ビットの並びが1010の場合に最も多く加減算を行うことになる。このとき，Rの下位５ビットは10100，11010，11101，11110と変化するため，加減算は３回実行することになる。したがって，空欄fは（ウ）が正解である。

・空欄g：プログラム２では，Nのビットを下位けたから順に調べ，１が現れたけた位置では減算を，次に１の並びが途切れて０が現れたけた位置で加算をする。N=7（0111）の場合，最下位けたの１で減算をし，続く２ビットの１では加減算は行わず，最上位けたの０が現れた際に加算をすることになる。例えば，M=3，N=7の場合はRの下位５ビットは01110となる。このとき，ビット番号１が１かつビット番号０が０なのでMの値（3）を減算してRを右に１ビット算術シフトする。シフト後のRの下位５ビットは00111となり，ビット番号１と０はいずれも１なので加減算は行わずにRをシフトする処理のみを行う。シフト後のRの下位５ビットは00011となるため，この場合もシフト演算のみ行い，Rの下位５ビットは00001となる。これでNの最上位けたの０がビット番号１まで移動した。また，Rを３回右に１ビット算術シフトしたので，Mを３ビット左にシフト（８倍）したことと同じことになり，最上位けたの０が現れた際の加算は，3×8=24を足すことになる。つまり，3×7=－3+24と計算されるのである。

これをM×($\boxed{\ \text{g}\ }$)の式に当てはめると，$3\times(-1+8)=M\times(-2^0+2^3)$と表すことができる。したがって，（ア）が正解である。