問8　数値計算と計算誤差（データ構造及びアルゴリズム）（H21秋-FE 午後問8）

【解答】
［設問］　a－イ，b－イ，c－エ，d－イ，e－エ

【解説】
ニュートン法を用いて方程式の解の一つを求めるアルゴリズムの問題である。方程式，ニュートン法という記述を見て戸惑った受験者も多かったと思われるが，この問題は必須問題であり，何とか得点につなげる必要がある。ニュートン法とは方程式の解を，任意に定めた解の予測値から始めて，計算を繰り返しながらその値を真の値に近づけていく方法である。問題文に方程式の解を求める手順は説明されており，ニュートン法の知識がなくても解答が可能となっている。テーマだけを見てあきらめることなく，根気良く取り組んで，空欄b，c，dは確実に得点したいものである。難易度は，やや難しいといえる。

〔プログラム1〕
(行番号)
```
1  ○主プログラム：プログラム1
2  ○整数型：i
3  ○実数型：d, f, x
4  ○実数型：a3, a2, a1, a0, b2, b1, b0

5   ・read(x, a3, a2, a1, a0)  /* x, a3～a0の値を読み込む。 */
6   ・b2 ← 3.0 × a3
7   ・b1 ← 2.0 × a2
8   ・b0 ← a1
9   ■ i：1, i ≦ 100, 1         /* 繰返し回数は100回とする。 */
10  │   ・f ← ((a3 × x + a2) × x + a1) × x + a0
11  │   ・d ←  (b2 × x + b1) × x + b0
12  │   ・print(x, f, d)        /* x, f, dの値を印字する。 */
13  │   ・x ← x － f ÷ d
14  ■
15 /* プログラム1の終わり */
```

プログラムと〔アルゴリズム1の説明〕にある処理手順を対応付けると次のようになる。
行番号5：手順(1)
行番号6～8：手順(2)
行番号10：手順(3)①
行番号11：手順(3)②
行番号12：手順(3)③
行番号13：手順(3)④

〔プログラム2〕
(行番号)
```
1  ○主プログラム：プログラム2
2  ○整数型：i, k, n          /* 1 ≦ n ≦ 9とする。 */
3  ○実数型：d, f, x
4  ○実数型：a[10], b[10]     /* 添字は0から始まる。 */

5   ・read(n, x)              /* n, xの値を読み込む。 */
6   ■ k：n, k ≧ 0, -1         /* kをn, n－1, …, 0として繰り返し, */
7   │   ・read(a[k])          /*  係数a[k]の値を順に読み込む。 */
8   ■
9   ■ k：n, k ≧ 1, -1         /* kをn, n－1, …, 1として繰り返す。 */
10  │   ・[ b ]
11  ■
12  ■ i：1, i ≦ 100, 1        /* 繰返し回数は100回とする。 */
13  │   ・f ← a[n] × x + a[n－1]
14  │   ・d ← [ c ]
15  │   ■ k：n－2, k ≧ 0, -1  /* kをn－2, n－3, …, 0として繰り返す。 */
16  │   │   ・f ← f × x + a[k]
17  │   │   ・d ← d × x + [ d ]
18  │   ■
19  │   ・print(x, f, d)      /* x, f, dの値を印字する。 */
20  │   ・x ← x － f ÷ d
21  ■
22 /* プログラム2の終わり */
```
※行番号9～11，13～18は，設問のプログラムの記述部分を反映

プログラムと〔アルゴリズム2の説明〕にある処理手順を対応付けると次のようになる。
行番号5～8：手順(1)
行番号9～11：手順(2)
行番号13，16：手順(3)①
行番号14，17：手順(3)②
行番号19：手順(3)③
行番号20：手順(3)④

最初に，実数値を扱う浮動小数点形式について整理する。コンピュータ内部では2進数で表現されるが，分かりやすいように10進数を用いる。

浮動小数点形式では，実数値を「±仮数×基数^{指数}」の形式で表す。例えば，123.45は$+1.2345\times10^{2}$のように表現される。“＋”が符号部，“1.2345”が仮数部，“10”が基数（10進数），“2”が指数部である。浮動小数点形式では，次の例のように実数値を正規化して表す。

$12345000000+1.2345\times10^{10}$
$-0.000054321-5.4321\times10^{-5}$

正規化とは，有効数字のけた数が同じになるよう，指数部の値を調整して仮数部の値を規定の範囲内に収める操作のことである。正規化を行うことで，限られたけた数でも非常に大きな数値や非常に小さな数値を扱うことが可能となる。有効数字とは，その目的に対して意味のある数字のことであり，有効数字のけた数を有効けた数という。表現するけた数が3けたの場合，0.012345は0.01としか表現できない。このとき，1より前の0には意味がないので，有効けた数は1けたとなる。そこで，1より前の0を排除して1.23×10^{-2}と正規化することで有効数字が3けた確保でき，数値の精度を上げることができる。ただし，浮動小数点形式の場合でも，有効けた数は有限であるため，すべての実数値を正しく表現できるとは限らない。また，コンピュータ内部では数値を2進数で表すため，10進数の実数値を2進数に基数変換した場合に，正確に表現できない値がある。10進数の0.1は2進数に変換すると，0.00011001100…となり，途中から循環小数になり有限けたで終わらない。そこで，最下位けたより小さい部分を四捨五入するなどの処理によって有効けた数に合わせる。このように，実数値の場合は，実際の値との差（誤差）が生じることがあり，実数値の演算では誤差への対処が必要となる。なお，有効けた数に合わせることによって発生する誤差を丸め誤差という。

［設問］
・空欄a：まず，プログラム1の印字結果（図1）の行番号6，7に印字された変数xの値（x_6，x_7とする）を算出した際に，変数fとdにどのような値が格納されていたのかを確認する。プログラム1を見ると，行番号5でxの初期値が読み込まれ，行番号10でfが，行番号11でdが算出されている。したがって，行番号12で最初に印字されるx，f，dの値（印字結果の1行目）はこれらの値を有効数字7けたで表示したものとなる。次に，行番号13でxの値を更新しているが，このとき計算に使用するx，f，dの値は，印字結果の1行目の印字に使用したx，f，dの値である。続いて，繰返しの先頭に戻り，行番号10でfを，行番号11でdを更新し，行番号12で印字結果の2行目のx，f，dを印字する。このことから，2行目以降に印字されたxの値は，1行前の印字に使用したx，f，dの値を用いて算出されていることが分かる。したがって，印字結果の6行目のx（x_6）の値は，5行目を印字した際のx，f，dの値，同様に7行目のx（x_7）の値は，6行目を印字した際のx，f，dの値から算出されたことになる。x_6，x_7の値はいずれも3.000000であるが，これは，有効数字7けたで表示するために，8けた目を四捨五入した結果であり，計算の結果が3であったということではない。x_6，x_7を算出する際のx，f，dの値は異なっており，また，この処理系の有効けた数は10進数で十数けた程度あるので，印字されているx，f，dの値も実際の変数の値とは異なっている。これらを考慮すると，演算結果のx_6とx_7は異なる値であると考えられる。したがって，(イ)が正解である。
・空欄b：〔プログラム2〕の行番号9～11は，〔アルゴリズム2の説明〕の手順(2)の処理である。設問の(1)で与えられた3次方程式$x^3-3x^2-x+3=0$を例に考えると，〔プログラム1〕の行番号6～8のb2に3.0×a3，b1に2.0×a2，b0にa1（1.0×a1）を格納する処理と同じである。a3，a2，a1の値は，行番号6～8の繰返し処理で配列aに格納されており，また，図2からも分かるようにb2，b1，b0の値は配列bのb[2]，b[1]，b[0]に格納されるので，配列a，bはそれぞれ次のようになる。

a	0	1	2	3	…
	3	−1	−3	1	…
	a_0の値	a_1の値	a_2の値	a_3の値	

b	0	1	2	…
	−1	−6	3	…
	b_0の値 1×a_1	b_1の値 2×a_2	b_2の値 3×a_3	

この場合，3次方程式であるためn＝3となり，繰返し処理の変数kの値は3，2，1と変化する。選択肢を見ると，配列bの添字はkを使って表しており，b[2]，b[1]，b[0]の順に格納するためには，b[k－1]とする必要がある。また，格納する値はb[2]が3×a[3]，b[1]が2×a[2]，b[0]が1×a[1]であり，kを使って表現するとk×a[k]となる。したがって，(イ)が正解である。
・空欄c，d：空欄bと同様に与えられた3次方程式を利用し，〔プログラム1〕の変数fを求める式「((a3×x＋a2)×x＋a1)×x＋a0」を〔プログラム2〕ではどのように変形されているのかを確認する。配列aには，空欄bで確認した値が格納されている。また，n＝3であるため，行番号13ではfの値は「a[3]×x＋a[2]＝a3×x＋a2」の部分になる。続いて行番号16では，k＝n－2＝1なので，「f×x＋a[1]＝(a3×x＋a2)×x＋a1」がfの値となる。k≧0の間繰り返すため再度行番号16が実行され，「f×x＋a[0]＝((a3×x＋a2)×x＋a1)×x＋a0」と処理されることになる。

これと同様に変数dを求める式を考える。〔プログラム1〕ではdを求める式は「(b2×x＋b1)×x＋b0」となっている（行番号11）。b2，b1，b0に対応する値は，〔プログラム2〕ではb[2]，b[1]，b[0]に格納されている。fの式を参考にすると，まず行番号14で「b2×x＋b1」の部分を処理すると考えられる。しかし，繰返し処理は2度実行されるため，行番号17を2度繰り返すと，「(b2×x＋b1)×x＋［ d ］」，「((b2×x＋b1)×x＋［ d ］)×x＋［ d ］」と処理され，「×x＋［ d ］」が一つ余分になる。そこで，行番号17の2度目の実行で「(b2×x＋b1)×x＋b0」という式になるように考えると，「(b2×x＋b1)」の部分が変数dとなり，空欄dの部分がb0（b[0]）になる。そうすると，行番号17の最初の実行の際の式は「d×x＋b1」となり，空欄dの部分がb1（b[1]）になる。そして，このとき変数dにはb2の値，つまりb[2]の値が格納されていれば正しい式となる。よって，空欄cの部分はb[2]を参照することになる。b[2]を参照するのは方程式が3次式であるためで，4次式の場合はb[3]となる。これを添字に変数を用いて表すにはb[n－1]とすればよい。また，空欄dでb[1]，b[0]の順に参照するには，b[k]と表せば正しく処理できる。したがって，空欄cは(エ)，空欄dは(イ)が正解である。

問8　数値計算と計算誤差（データ構造及びアルゴリズム）（H21秋-FE 午後問8）

【解答】
［設問］　a－イ，b－イ，c－エ，d－イ，e－エ

【解説】
ニュートン法を用いて方程式の解の一つを求めるアルゴリズムの問題である。方程式，ニュートン法という記述を見て戸惑った受験者も多かったと思われるが，この問題は必須問題であり，何とか得点につなげる必要がある。ニュートン法とは方程式の解を，任意に定めた解の予測値から始めて，計算を繰り返しながらその値を真の値に近づけていく方法である。問題文に方程式の解を求める手順は説明されており，ニュートン法の知識がなくても解答が可能となっている。テーマだけを見てあきらめることなく，根気良く取り組んで，空欄b，c，dは確実に得点したいものである。難易度は，やや難しいといえる。

〔プログラム1〕
(行番号)
```
1  ○主プログラム：プログラム1
2  ○整数型：i
3  ○実数型：d, f, x
4  ○実数型：a3, a2, a1, a0, b2, b1, b0

5   ・read(x, a3, a2, a1, a0)  /* x, a3～a0の値を読み込む。 */
6   ・b2 ← 3.0 × a3
7   ・b1 ← 2.0 × a2
8   ・b0 ← a1
9   ■ i：1, i ≦ 100, 1         /* 繰返し回数は100回とする。 */
10  │   ・f ← ((a3 × x + a2) × x + a1) × x + a0
11  │   ・d ←  (b2 × x + b1) × x + b0
12  │   ・print(x, f, d)        /* x, f, dの値を印字する。 */
13  │   ・x ← x － f ÷ d
14  ■
15 /* プログラム1の終わり */
```

プログラムと〔アルゴリズム1の説明〕にある処理手順を対応付けると次のようになる。
行番号5：手順(1)
行番号6～8：手順(2)
行番号10：手順(3)①
行番号11：手順(3)②
行番号12：手順(3)③
行番号13：手順(3)④

〔プログラム2〕
(行番号)
```
1  ○主プログラム：プログラム2
2  ○整数型：i, k, n          /* 1 ≦ n ≦ 9とする。 */
3  ○実数型：d, f, x
4  ○実数型：a[10], b[10]     /* 添字は0から始まる。 */

5   ・read(n, x)              /* n, xの値を読み込む。 */
6   ■ k：n, k ≧ 0, -1         /* kをn, n－1, …, 0として繰り返し, */
7   │   ・read(a[k])          /*  係数a[k]の値を順に読み込む。 */
8   ■
9   ■ k：n, k ≧ 1, -1         /* kをn, n－1, …, 1として繰り返す。 */
10  │   ・[ b ]
11  ■
12  ■ i：1, i ≦ 100, 1        /* 繰返し回数は100回とする。 */
13  │   ・f ← a[n] × x + a[n－1]
14  │   ・d ← [ c ]
15  │   ■ k：n－2, k ≧ 0, -1  /* kをn－2, n－3, …, 0として繰り返す。 */
16  │   │   ・f ← f × x + a[k]
17  │   │   ・d ← d × x + [ d ]
18  │   ■
19  │   ・print(x, f, d)      /* x, f, dの値を印字する。 */
20  │   ・x ← x － f ÷ d
21  ■
22 /* プログラム2の終わり */
```
※行番号9～11，13～18は，設問のプログラムの記述部分を反映

プログラムと〔アルゴリズム2の説明〕にある処理手順を対応付けると次のようになる。
行番号5～8：手順(1)
行番号9～11：手順(2)
行番号13，16：手順(3)①
行番号14，17：手順(3)②
行番号19：手順(3)③
行番号20：手順(3)④

最初に，実数値を扱う浮動小数点形式について整理する。コンピュータ内部では2進数で表現されるが，分かりやすいように10進数を用いる。

浮動小数点形式では，実数値を「±仮数×基数^{指数}」の形式で表す。例えば，123.45は$+1.2345\times10^{2}$のように表現される。“＋”が符号部，“1.2345”が仮数部，“10”が基数（10進数），“2”が指数部である。浮動小数点形式では，次の例のように実数値を正規化して表す。

$12345000000+1.2345\times10^{10}$
$-0.000054321-5.4321\times10^{-5}$

正規化とは，有効数字のけた数が同じになるよう，指数部の値を調整して仮数部の値を規定の範囲内に収める操作のことである。正規化を行うことで，限られたけた数でも非常に大きな数値や非常に小さな数値を扱うことが可能となる。有効数字とは，その目的に対して意味のある数字のことであり，有効数字のけた数を有効けた数という。表現するけた数が3けたの場合，0.012345は0.01としか表現できない。このとき，1より前の0には意味がないので，有効けた数は1けたとなる。そこで，1より前の0を排除して1.23×10^{-2}と正規化することで有効数字が3けた確保でき，数値の精度を上げることができる。ただし，浮動小数点形式の場合でも，有効けた数は有限であるため，すべての実数値を正しく表現できるとは限らない。また，コンピュータ内部では数値を2進数で表すため，10進数の実数値を2進数に基数変換した場合に，正確に表現できない値がある。10進数の0.1は2進数に変換すると，0.00011001100…となり，途中から循環小数になり有限けたで終わらない。そこで，最下位けたより小さい部分を四捨五入するなどの処理によって有効けた数に合わせる。このように，実数値の場合は，実際の値との差（誤差）が生じることがあり，実数値の演算では誤差への対処が必要となる。なお，有効けた数に合わせることによって発生する誤差を丸め誤差という。

［設問］
・空欄a：まず，プログラム1の印字結果（図1）の行番号6，7に印字された変数xの値（x_6，x_7とする）を算出した際に，変数fとdにどのような値が格納されていたのかを確認する。プログラム1を見ると，行番号5でxの初期値が読み込まれ，行番号10でfが，行番号11でdが算出されている。したがって，行番号12で最初に印字されるx，f，dの値（印字結果の1行目）はこれらの値を有効数字7けたで表示したものとなる。次に，行番号13でxの値を更新しているが，このとき計算に使用するx，f，dの値は，印字結果の1行目の印字に使用したx，f，dの値である。続いて，繰返しの先頭に戻り，行番号10でfを，行番号11でdを更新し，行番号12で印字結果の2行目のx，f，dを印字する。このことから，2行目以降に印字されたxの値は，1行前の印字に使用したx，f，dの値を用いて算出されていることが分かる。したがって，印字結果の6行目のx（x_6）の値は，5行目を印字した際のx，f，dの値，同様に7行目のx（x_7）の値は，6行目を印字した際のx，f，dの値から算出されたことになる。x_6，x_7の値はいずれも3.000000であるが，これは，有効数字7けたで表示するために，8けた目を四捨五入した結果であり，計算の結果が3であったということではない。x_6，x_7を算出する際のx，f，dの値は異なっており，また，この処理系の有効けた数は10進数で十数けた程度あるので，印字されているx，f，dの値も実際の変数の値とは異なっている。これらを考慮すると，演算結果のx_6とx_7は異なる値であると考えられる。したがって，(イ)が正解である。
・空欄b：〔プログラム2〕の行番号9～11は，〔アルゴリズム2の説明〕の手順(2)の処理である。設問の(1)で与えられた3次方程式$x^3-3x^2-x+3=0$を例に考えると，〔プログラム1〕の行番号6～8のb2に3.0×a3，b1に2.0×a2，b0にa1（1.0×a1）を格納する処理と同じである。a3，a2，a1の値は，行番号6～8の繰返し処理で配列aに格納されており，また，図2からも分かるようにb2，b1，b0の値は配列bのb[2]，b[1]，b[0]に格納されるので，配列a，bはそれぞれ次のようになる。

a	0	1	2	3	…
	3	−1	−3	1	…
	a_0の値	a_1の値	a_2の値	a_3の値	

b	0	1	2	…
	−1	−6	3	…
	b_0の値 1×a_1	b_1の値 2×a_2	b_2の値 3×a_3	

この場合，3次方程式であるためn＝3となり，繰返し処理の変数kの値は3，2，1と変化する。選択肢を見ると，配列bの添字はkを使って表しており，b[2]，b[1]，b[0]の順に格納するためには，b[k－1]とする必要がある。また，格納する値はb[2]が3×a[3]，b[1]が2×a[2]，b[0]が1×a[1]であり，kを使って表現するとk×a[k]となる。したがって，(イ)が正解である。
・空欄c，d：空欄bと同様に与えられた3次方程式を利用し，〔プログラム1〕の変数fを求める式「((a3×x＋a2)×x＋a1)×x＋a0」を〔プログラム2〕ではどのように変形されているのかを確認する。配列aには，空欄bで確認した値が格納されている。また，n＝3であるため，行番号13ではfの値は「a[3]×x＋a[2]＝a3×x＋a2」の部分になる。続いて行番号16では，k＝n－2＝1なので，「f×x＋a[1]＝(a3×x＋a2)×x＋a1」がfの値となる。k≧0の間繰り返すため再度行番号16が実行され，「f×x＋a[0]＝((a3×x＋a2)×x＋a1)×x＋a0」と処理されることになる。

これと同様に変数dを求める式を考える。〔プログラム1〕ではdを求める式は「(b2×x＋b1)×x＋b0」となっている（行番号11）。b2，b1，b0に対応する値は，〔プログラム2〕ではb[2]，b[1]，b[0]に格納されている。fの式を参考にすると，まず行番号14で「b2×x＋b1」の部分を処理すると考えられる。しかし，繰返し処理は2度実行されるため，行番号17を2度繰り返すと，「(b2×x＋b1)×x＋［ d ］」，「((b2×x＋b1)×x＋［ d ］)×x＋［ d ］」と処理され，「×x＋［ d ］」が一つ余分になる。そこで，行番号17の2度目の実行で「(b2×x＋b1)×x＋b0」という式になるように考えると，「(b2×x＋b1)」の部分が変数dとなり，空欄dの部分がb0（b[0]）になる。そうすると，行番号17の最初の実行の際の式は「d×x＋b1」となり，空欄dの部分がb1（b[1]）になる。そして，このとき変数dにはb2の値，つまりb[2]の値が格納されていれば正しい式となる。よって，空欄cの部分はb[2]を参照することになる。b[2]を参照するのは方程式が3次式であるためで，4次式の場合はb[3]となる。これを添字に変数を用いて表すにはb[n－1]とすればよい。また，空欄dでb[1]，b[0]の順に参照するには，b[k]と表せば正しく処理できる。したがって，空欄cは(エ)，空欄dは(イ)が正解である。