

次の問8は必須問題です。必ず解答してください。

問8 次のアルゴリズムの説明及びプログラムを読んで、設問に答えよ。

方程式の解の一つを求めるアルゴリズムである。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。この方法は、ニュートン法と呼ばれる。

[アルゴリズム1の説明]

3次方程式 $a_3x^3 + a_2x^2 + a_1x + a_0 = 0$ の解の一つを、次の手順で求める。

- (1) 解の予測値 x ，係数 a_3 ， a_2 ， a_1 ， a_0 を読み込む。
 - (2) $3 \times a_3$ の値を b_2 に， $2 \times a_2$ の値を b_1 に， $1 \times a_1$ の値を b_0 に，それぞれ求める。
 - (3) 次の①～④の処理を一定の回数繰り返す。
 - ① $a_3x^3 + a_2x^2 + a_1x + a_0$ の値を求め，これを f とする。
 - ② $b_2x^2 + b_1x + b_0$ の値を求め，これを d とする。
 - ③ x ， f ， d の値を印字する。
 - ④ $x - \frac{f}{d}$ の値（解の一つにより近い値となる）を求め，これを新たな x とする。
- プログラム1は，このアルゴリズム1を実装したものである。

[アルゴリズム2の説明]

アルゴリズム1を一般化して， n 次方程式 $a_nx^n + a_{n-1}x^{n-1} + \dots + a_1x + a_0 = 0$ の解の一つを，次の手順で求める。

なお，方程式によっては解が求められない場合がある。

- (1) 次数 n ，解の予測値 x ，係数 a_n ， a_{n-1} ， \dots ， a_1 ， a_0 を読み込む。
 - (2) $n \times a_n$ の値を b_{n-1} に， $(n-1) \times a_{n-1}$ の値を b_{n-2} に， \dots ， $2 \times a_2$ の値を b_1 に， $1 \times a_1$ の値を b_0 に，それぞれ求める。
 - (3) 次の①～④の処理を一定の回数繰り返す。
 - ① $a_nx^n + a_{n-1}x^{n-1} + \dots + a_1x + a_0$ の値を求め，これを f とする。
 - ② $b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \dots + b_1x + b_0$ の値を求め，これを d とする。
 - ③ x ， f ， d の値を印字する。
 - ④ $x - \frac{f}{d}$ の値（解の一つにより近い値となる）を求め，これを新たな x とする。
- プログラム2は，このアルゴリズム2を実装したものである。

[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 として繰返し, */
7  |   ・read(a[k])      /* 係数 a[k] の値を順に読み込む。 */
8  | ■
9  | ┌───────────┐
10 | | 手順(2)の処理 |
11 | | ───────────┘ |
12 | ■ i: 1, i ≤ 100, 1 /* 繰返し回数は 100 回とする。 */
13 | | ┌───────────┐
14 | | | 手順(3)の①と②の処理 |
15 | | | ───────────┘ |
16 | | |
17 | | |
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 */

```

設問 次の記述中の に入れる正しい答えを、解答群の中から選べ。

- (1) 解の予測値 $x=2.5$ 、係数 $a_3=1$ 、 $a_2=-3$ 、 $a_1=-1$ 、 $a_0=3$ を与えて、3次方程式 $x^3 - 3x^2 - x + 3 = 0$ の解の一つを求める（解は 3, 1, -1）。プログラム 1 がある処理系で実行した結果、図 1 に示すとおり解の一つである $x=3$ が近似的に得られた。

(行番号)	x	f	d	
1	2.500000	-2.625000	2.750000	注1 数値の後の(-k)は、 $\times 10^{-k}$ を示す。例えば、 5.548452(-2)は、 5.548452×10^{-2} 、すなわち 0.05548452を表す。
2	3.454545	4.969947	14.07438	
3	3.101425	8.741682(-1)	9.247965	
4	3.006900	5.548452(-2)	8.082941	
5	3.000035	2.833717(-4)	8.000425	
6	3.000000	7.527369(-9)	8.000000	注2 表示は有効数字7けた (8けた目を四捨五入)
7	3.000000	0.000000	8.000000	
8	3.000000	0.000000	8.000000	

図1 プログラム1の印字結果

この印字結果の行番号6, 7の x の値（網掛けの部分）はいずれも3.000000である。行番号6, 7を印字した時点で変数 x に保持されていた実際の値をそれぞれ x_6 , x_7 で表すと、 a

なお、この処理系では、実数型は2進数の浮動小数点形式であって、有効けた数は10進数で十数けた程度であることが分かっている。

- (2) プログラム2では、係数 a_k ($k:n, n-1, \dots, 1, 0$)の値を配列aの要素 $a[k]$ に、 b_k ($k:n-1, n-2, \dots, 1, 0$)の値を配列bの要素 $b[k]$ に、それぞれ図2のように格納している。

要素番号	0	1	2	...	n-1	n	...
配列 a	a_0	a_1	a_2	...	a_{n-1}	a_n	...
要素番号	0	1	...	n-3	n-2	n-1	...
配列 b	$1 \times a_1$	$2 \times a_2$...	$(n-2) \times a_{n-2}$	$(n-1) \times a_{n-1}$	$n \times a_n$...

図2 係数 a_k , b_k の値の格納

プログラム2の行番号9～11は、アルゴリズム2の手順(2)の処理である。この部分のプログラムは、次のようになる。

[プログラム2の一部]

(行番号)

```

9   ■ k: n, k ≥ 1, -1 /* kをn, n-1, ..., 1として繰り返す。 */
10   |   · b
11   ■

```

また、行番号13～18は、アルゴリズム2の手順(3)の①と②の処理である。プログラム1では、例えば f の値 $a_3x^3 + a_2x^2 + a_1x + a_0$ を求める式を、

$$f \leftarrow ((a_3 \times x + a_2) \times x + a_1) \times x + a_0$$

と変形して、演算回数を減らす工夫をしている。この部分にも同様の工夫をすると、プログラムは次のようになる。

[プログラム2の一部]

(行番号)

```

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   ■

```

(3) 次数 $n=4$ 、係数 $a_4=1$ 、 $a_3=-8$ 、 $a_2=24$ 、 $a_1=-32$ 、 $a_0=16$ として、4次方程式 $x^4 - 8x^3 + 24x^2 - 32x + 16 = 0$ の解を求める(4個の解がすべて2)。解の予測値を $x=2.00001$ として、ある処理系でプログラム2を実行したところ、図3に示すとおり印字結果となった。

(行番号)	x	f	d
1	2.000010	-3.552714(-15)	3.996803(-15)
2	2.888899	6.243232(-1)	2.809423
3	2.666674	1.975398(-1)	1.185225
4	2.500006	6.250281(-2)	5.000169(-1)
5	2.375004	1.977628(-2)	2.109446(-1)

図3 プログラム2の印字結果

この印字結果の行番号2では、 x の値（網掛けの部分）が解である2から遠ざかってしまっている。その原因を調べるため、 f を求める式に実際の数値を当てはめて、

$$\underbrace{(((1.0 \times 2.00001 - 8.0) \times 2.00001 + 24.0) \times 2.00001 - 32.0) \times 2.00001}_{(A)} + 16.0$$

として、(A)の部分の中間結果を印字するプログラムを作り、同じ処理系で実行した。印字結果は-16.000000であり、正確な値-15.999999999999999999999999と有効数字7けたで一致した。しかし、行番号1で印字された f の値は、正確な値である 10^{-20} （印字の表記では1.000000(-20)）とは異なっている。

これらのことから判断して、(A)の部分では演算の過程で e が徐々に累積し、(A)の計算結果に16.0を加算するときに、けた落ちが発生したと考えられる。

aに関する解答群

ア $x_6 = x_7$ である

イ $x_6 \neq x_7$ である

ウ $x_6 = x_7$ とも $x_6 \neq x_7$ ともいえない

bに関する解答群

ア $b[k-1] \leftarrow (k-1) \times a[k]$

イ $b[k-1] \leftarrow k \times a[k]$

ウ $b[k] \leftarrow k \times a[k+1]$

エ $b[k] \leftarrow (k+1) \times a[k+1]$

c, dに関する解答群

ア $b[k-1]$

ウ $b[k+1]$

オ $b[n-1] \times x$

イ $b[k]$

エ $b[n-1]$

カ $b[n-1] \times x + b[n-2]$

eに関する解答群

ア けたあふれ

ウ 指数下位けたあふれ

イ けた落ち

エ 丸め誤差