

数値解析 課題 配列

行列（2次元配列）の要素を様々な並べ替えるプログラムを作成した。下に示す C 言語プログラムと実行例を読んで、以下の各問に答えよ。

```
#include <stdio.h>

#define DIM      4

void init(double x[DIM][DIM])
{
    int i, j;

    for (i = 0; i < DIM; i++)
        for (j = 0; j < DIM; j++)
            x[i][j] = i + ① + 1.1;
}

void dump(double x[DIM][DIM])
{
    int i, j;

    for (i = 0; i < DIM; i++) {
        for (j = 0; j < DIM; j++)
            printf("%8.1f ", x[i][j]);
        printf("\n");
    }
    printf("\n");
}

void func1(double x[DIM][DIM])
{
    int i, j;
    double t;

    for (i = 0; 2*i < DIM; i++) {
        for (j = 0; j < DIM; j++) {
            t = x[i][j];
            x[i][j] = x[DIM-i-1][j];
            x[DIM-i-1][j] = t;
        }
    }
}

void func2(double x[DIM][DIM])
{
    int i, j;
    double t;

    for (j = 0; 2*j < DIM; j++) {
        for (i = 0; i < DIM; i++) {
            t = x[i][j];
            x[i][j] = x[i][DIM-j-1];
            x[i][DIM-j-1] = t;
        }
    }
}
```

```
void func3(double x[DIM][DIM])
{
    int i, j;
    double t;

    for (i = 0; i < DIM; i++)
        for (j = i+1; j < DIM; j++) {
            t = x[i][j];
            x[i][j] = x[j][i];
            x[j][i] = t;
        }
}

int main()
{
    double a[DIM][DIM];

    init(a);
    dump(a);      // ②
    func1(a);
    dump(a);      // ③
    func2(a);
    dump(a);      // ④

    init(a);
    func3(a);
    dump(a);      // ⑤

    return 0;
}
```

- (1) 関数 **main** の②を実行したとき、関数 **dump** の出力は以下通りだったとする。
関数 **init** の空欄 ① を埋めるのに最も適切な記述を選択肢から選んで記号で答えよ。

1.1	1.2	1.3	1.4
2.1	2.2	2.3	2.4
3.1	3.2	3.3	3.4
4.1	4.2	4.3	4.4

イ. j ロ. $j-1$ ハ. $j*10$ ニ. $j/10$ ホ. $j*0.1$

- (2) プログラム中の③で、関数 **dump** の出力はどのようなになるか、選択肢から選んで答えよ。
(3) プログラム中の④で、関数 **dump** の出力はどのようなになるか、選択肢から選んで答えよ。
(4) プログラム中の⑤で、関数 **dump** の出力はどのようなになるか、選択肢から選んで答えよ。

選択肢

イ.

1.1	1.2	1.3	1.4
2.1	2.2	2.3	2.4
3.1	3.2	3.3	3.4
4.1	4.2	4.3	4.4

ロ.

1.1	2.1	3.1	4.1
1.2	2.2	3.2	4.2
1.3	2.3	3.3	4.3
1.4	2.4	3.4	4.4

ハ.

4.1	4.2	4.3	4.4
3.1	3.2	3.3	3.4
2.1	2.2	2.3	2.4
1.1	1.2	1.3	1.4

ニ.

4.4	4.3	4.2	4.1
3.4	3.3	3.2	3.1
2.4	2.3	2.2	2.1
1.4	1.3	1.2	1.1