プログラミング演習I 課題ドキュメント

b162392

構築: Doxygen 1.8.6

2016 年 10 月 26 日 (水) 21 時 24 分 44 秒

Contents

1 ファイル詳解

1.1 main.c ファイル

2 次元（または 3 次元）行列の行列式を計算する

```
#include <stdio.h>
```

関数

- int main (void)
  2 次元（または 3 次元）行列の行列式を計算する

1.1.1 詳解

2 次元（または 3 次元）行列の行列式を計算する

日付

2016/10/26

著者

佐伯雄飛
B162392

main.c に定義があります。

1.1.2 関数詳解

1.1.2.1 int main (void)

2 次元（または 3 次元）行列の行列式を計算する
入力：

- 標準入力には，まず行列が 2 次元か 3 次元かを区別する整数が与えられる． 2 次元なら 2，3 次元なら 3.

もしそれ以外の数値が与えられたら，プログラムは何も表示せずに終了する（return 0 で）.

- それに引き続いて，2 次元行列の要素 ($a_{11}, a_{12}, a_{21}, a_{22}$) または 3 次元行列の要素 ($a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23}, a_{31}, a_{32}, a_{33}$) がこの順番で与えられる． 値は実数.

出力：

- 与えられた行列の行列式を計算し，それを標準出力に表示する．
- 数値は小数点第 5 位まで表示する（%.5f）．

入力例：

```
2
0.933942777791 0.402727471670
0.181429362962 0.264556070201
```

出力例：

```
0.17401
```

入力例：

```
3
0.638833848053 0.0150411981842 0.402224237664
0.911355582360 0.8907163041810 0.419912154151
0.383181998496 0.2191199720820 0.678895763639
```

出力例：

```
0.26368
```

日付

2016/10/26

著者

佐伯雄飛，B162392

main.c の 行目に定義があります。

```
00055                {
00056   int n;
00057   scanf("%d", &n);
00058   if (n == 2 || n == 3) {
00059     float A[n][n];  // matrix A
00060
00061     for (int i = 0; i < n; i++) {
00062       for (int j = 0; j < n; j++) {
00063         scanf("%f", &A[i][j]);
00064       }
00065     }
00066
00067     float det;
00068     if (n == 2) {
00069       det = A[0][0] * A[1][1] - A[1][0] * A[0][1];  // a_11 a_22 - a_21 a_12
00070     } else {
00071       // n == 3 の場合のコードをここに書いてください
00072       det = A[0][0] * A[1][1] * A[2][2] - A[0][2] * A[1][1] * A[2][0] -
00073             A[1][0] * A[0][1] * A[2][2] + A[1][0] * A[2][1] * A[0][2] +
00074             A[0][1] * A[1][2] * A[2][0] - A[0][0] * A[1][2] * A[2][1];
00075     }
00076     printf("%.5f\n", det);
00077   } else {
00078     printf("");
00079   }
00080   return 0;
00081 }
```

1.2 main.c

```
00001 /**  @file main.c
00002  *   @brief  2 次元（または 3 次元）行列の行列式を計算する
00003  *   @date   2016/10/26
00004  *   @author 佐伯雄飛
00005  *   @author B162392
00006  */
00007
00008 #include <stdio.h>
00009
00010 /** @fn int main(void)
00011  *  @brief  2 次元（または 3 次元）行列の行列式を計算する
00012  *
00013  *  入力：
00014  *  - 標準入力には，まず行列が 2 次元か 3 次元かを区別する整数が与えられる．
00015  *    2 次元なら 2，3 次元なら 3．
00016  *
00017 もしそれ以外の数値が与えられたら，プログラムは何も表示せずに終了する（return
00018 0 で）．
00019  *  - それに引き続いて，2 次元行列の要素
00020  *    (\f$ a_{11},a_{12},a_{21},a_{22} \f$)
00021  *    または 3 次元行列の要素
00022  *    (\f$ a_{11},a_{12},a_{13},a_{21},a_{22},a_{23},a_{31},a_{32},a_{33}\f$)
00023  *    がこの順番で与えられる．
00024  *    値は実数．
00025  *
00026  *  出力：
00027  *  - 与えられた行列の行列式を計算し，それを標準出力に表示する．
00028  *  - 数値は小数点第 5 位まで表示する（%.5f）．
00029  *
00030  *  入力例：
00031 \verbatim
00032 2
00033 0.933942777791 0.402727471670
00034 0.181429362962 0.264556070201
00035 \endverbatim
00036   *  出力例：
00037 \verbatim
00038 0.17401
00039 \endverbatim
00040   *  入力例：
00041 \verbatim
00042 3
00043 0.638833848053 0.0150411981842 0.402224237664
00044 0.911355582360 0.8907163041810 0.419912154151
00045 0.383181998496 0.2191199720820 0.678895763639
00046 \endverbatim
00047   *  出力例：
00048 \verbatim
00049 0.26368
00050 \endverbatim
00051  *  @date   2016/10/26
00052  *  @author 佐伯雄飛, B162392
00053  */
00054
00055 int main(void) {
00056   int n;
00057   scanf("%d", &n);
00058   if (n == 2 || n == 3) {
00059     float A[n][n];  // matrix A
00060
00061     for (int i = 0; i < n; i++) {
00062       for (int j = 0; j < n; j++) {
00063         scanf("%f", &A[i][j]);
00064       }
00065     }
00066
00067     float det;
00068     if (n == 2) {
00069       det = A[0][0] * A[1][1] - A[1][0] * A[0][1];  // a_11 a_22 - a_21 a_12
00070     } else {
00071       // n == 3 の場合のコードをここに書いてください
00072       det = A[0][0] * A[1][1] * A[2][2] - A[0][2] * A[1][1] * A[2][0] -
00073             A[1][0] * A[0][1] * A[2][2] + A[1][0] * A[2][1] * A[0][2] +
00074             A[0][1] * A[1][2] * A[2][0] - A[0][0] * A[1][2] * A[2][1];
00075     }
00076     printf("%.5f\n", det);
00077   } else {
00078     printf("");
00079   }
00080   return 0;
00081 }
```