

# 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 27 日 (木) 16 時 55 分 25 秒

## Contents

1	ファイル詳解	1
1.1	main.c ファイル	1
1.1.1	詳解	1
1.1.2	関数詳解	1
1.2	main.c	2

## 1 ファイル詳解

### 1.1 main.c ファイル

2 つの静電荷間に働く力の計算

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

関数

- int `main` (void)  
2 つの静電荷間に働く力の計算

#### 1.1.1 詳解

2 つの静電荷間に働く力の計算

日付

2016/10/27

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 つの静電荷間に働く力の計算

標準入力から 2 つの静電荷の電荷量と 3 次元座標を読み込み，その間に働く力をクーロンの法則により求める

入力：

- 標準入力の最初の実数は，1 つ目の静電荷の電荷量
- それに引き続く 3 個の実数は 1 つ目の静電荷の座標 (x,y,z)
- 次に続く実数は，2 つ目の静電荷の電荷量

- それに引き続く 3 個の実数は 2 つ目の静電荷の座標 (x,y,z)

出力 :

- 2 つの静電荷間に働く力を計算し , 表示する .
- 数値は指数形式で , 小数点第 6 位まで表示する (%.6e) .

入力例 :

```
0.000308699789318
0.000402224300987 0.000279583777408 0.000174189877294
-0.000239153024041
-5.80356914685e-05 -0.000160431721634 -6.19990827336e-05
```

出力例 :

```
1.440554e+09
```

入力例 :

```
5.89869864357e-05
-0.000307976825127 -0.000255850688569 -0.000440580236465
-0.000381973244918
-0.000129578394884 -0.000480165202626 -0.000181748151379
```

出力例 :

```
-1.359709e+09
```

```
@date 2016/10/27
@author 佐伯雄飛 , B162392
```

[main.c](#) の行目に定義があります。

```
00053     {
00054         float k0 = 9.0e9;           // about 9.0 x 10^9 [N m^2 / C^2]
00055         float q[2];                // charges of two particles [C]
00056         float r[2][3];             // 3D coordinates of two particles [m]
00057         float distance = 0.0;      // (square of) distance between two particles
00058         float tempt;
00059         for (int i = 0; i < 2; i++) {
00060             scanf("%e", &q[i]);    // ここで q を読み込む
00061             for (int j = 0; j < 3; j++) {
00062                 scanf("%e", &r[i][j]); // ここで r を読み込む
00063             }
00064         }
00065
00066         for (int j = 0; j < 3; j++) {
00067             float temp = r[0][j] - r[1][j];
00068             distance += temp * temp;
00069         }
00070         // ここで distance を計算する
00071
00072         float F;
00073         if (distance > 0) {
00074             F = k0 * q[0] * q[1] / distance; // [N]
00075         } else {
00076             F = 0;
00077         }
00078
00079         printf("%.6e", F);
00080         printf("\n");
00081         // ここで F を表示する
00082
00083         return 0;
00084     }
```

## 1.2 main.c

```

00001 /** @file main.c
00002  * @brief 2つの静電荷間に働く力の計算
00003  * @date 2016/10/27
00004  * @author 佐伯雄飛
00005  * @author B162392
00006 */
00007
00008 #include <stdio.h>
00009
00010 /** @fn int main(void)
00011  * @brief 2つの静電荷間に働く力の計算
00012  *
00013  * 標準入力から2つの静電荷の電荷量と3次元座標を読み込み、
00014  * その間に働く力をクーロンの法則により求める
00015  *
00016  * 入力：
00017  * - 標準入力の最初の実数は、1つ目の静電荷の電荷量
00018  * - それに引き続く3個の実数は1つ目の静電荷の座標 (x, y, z)
00019  * - 次に続く実数は、2つ目の静電荷の電荷量
00020  * - それに引き続く3個の実数は2つ目の静電荷の座標 (x, y, z)
00021  *
00022  * 出力：
00023  * - 2つの静電荷間に働く力を計算し、表示する。
00024  * - 数値は指数形式で、小数点第6位まで表示する (%.6e)。
00025  *
00026  * 入力例：
00027 \verbatim
00028 0.000308699789318
00029 0.000402224300987 0.000279583777408 0.000174189877294
00030 -0.000239153024041
00031 -5.80356914685e-05 -0.000160431721634 -6.19990827336e-05
00032 \endverbatim
00033 * 出力例：
00034 \verbatim
00035 1.440554e+09
00036 \endverbatim
00037 * 入力例：
00038 \verbatim
00039 5.89869864357e-05
00040 -0.000307976825127 -0.000255850688569 -0.000440580236465
00041 -0.000381973244918
00042 -0.000129578394884 -0.000480165202626 -0.000181748151379
00043 \endverbatim
00044 * 出力例：
00045 \verbatim
00046 -1.359709e+09
00047 \endverbatim
00048 *
00049 * @date 2016/10/27
00050 * @author 佐伯雄飛, B162392
00051 */
00052
00053 int main(void) {
00054     float k0 = 9.0e9; // about 9.0 x 10^9 [N m^2 / C^2]
00055     float q[2]; // charges of two particles [C]
00056     float r[2][3]; // 3D coordinates of two particles [m]
00057     float distance = 0.0; // (square of) distance between two particles
00058     float tempt;
00059     for (int i = 0; i < 2; i++) {
00060         scanf("%e", &q[i]); // ここで q を読み込む
00061         for (int j = 0; j < 3; j++) {
00062             scanf("%e", &r[i][j]); // ここで r を読み込む
00063         }
00064     }
00065
00066     for (int j = 0; j < 3; j++) {
00067         float temp = r[0][j] - r[1][j];
00068         distance += temp * temp;
00069     }
00070     // ここで distance を計算する
00071
00072     float F;
00073     if (distance > 0) {
00074         F = k0 * q[0] * q[1] / distance; // [N]
00075     } else {
00076         F = 0;
00077     }
00078
00079     printf("%.6e", F);
00080     printf("\n");
00081     // ここで F を表示する
00082
00083     return 0;
00084 }

```