

# 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 03 分 50 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	3

## 1 ファイル詳解

### 1.1 main.c ファイル

数式微分を行う

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

関数

- int `main` (void)  
数式微分を行う

#### 1.1.1 詳解

数式微分を行う

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 )

数式微分を行う

標準入力から  $n$  次多項式の係数を読み込み，その多項式と，微分した  $n-1$  次多項式を表示する  
入力：

- 標準入力の最初の整数は，多項式の次数  $n$  ( $n$  は 0 以上の整数)
- それに引き続く  $n$  個の実数は  $n$  次多項式の係数

出力：

- $n$  次多項式を以下のフォーマットで出力  $c_0+c_1x+c_2x^2+\dots+c_nx^n$
- 微分した  $n-1$  次多項式を以下のフォーマットで出力  $c_0+c_1x+c_2x^2+\dots+c_nx^{n-1}$
- 空白は表示しない
- 係数が負であれば, +記号を-にする
- 定数係数 ( $c_0$ ) が負であれば-記号を表示する. 正であれば記号を表示しない
- 係数は小数点第 1 位まで表示する (%.1f).
- 次数  $n$  は, 2 次以降の項は整数だけを表示し  $x^n$  とする. 1 次の項は  $x$  と表示する. 定数項は  $x$  を表示しない.

注意:

- 次数  $n$  が負の場合には, 何も表示せずに終了する (return 0 で)
- 次数  $n$  が 0 の場合は定数項だけであり, 微分したら 0 を表示する

入力例:

```
5 0.299015168361 -0.634498680199 0.076517409876 -0.155687714204 0.50453614886
0.260700633183
```

出力例:

```
0.3-0.6x+0.1x^2-0.2x^3+0.5x^4+0.3x^5
-0.6+0.2x-0.5x^2+2.0x^3+1.3x^4
```

入力例:

```
-1
```

出力例:

入力例:

```
0 0.338757150218
```

出力例:

```
0.3
0.0
```

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。

```
00065     {
00066     int n; // order of polynomial
00067     scanf("%d", &n);
00068     n++;
00069
00070     if (n < 1) {
00071         printf("");
00072     }
00073
00074     else {
00075         float c[n]; // coefficient of n-th order term
00076         for (int i = 0; i < n; i++) {
00077             scanf("%f", &c[i]);
00078         }
00079     }
```

```

00080 // print the polynomial: c0 + c1 x + c2 x^2 + .... + cn x^n
00081
00082 for (int i = 0; i < n; i++) {
00083     printf(i > 1 ? "%+.1fx^%d" : (i == 0 ? "%.1f" : "%+.1fx"), c[i], i);
00084 }
00085 printf("\n");
00086
00087 if (n == 1) {
00088     printf("0.0");
00089     printf("\n");
00090 }
00091
00092 else {
00093     // differentiation
00094     for (int i = 1; i < n; i++) {
00095         c[i] *= i;
00096         c[i - 1] = c[i];
00097     }
00098
00099     // print the derivative: c0' + c1' x + c2' x^2 + .... + cn' x^n-1
00100     for (int i = 0; i < n - 1; i++) {
00101         printf(i > 1 ? "%+.1fx^%d" : (i == 0 ? "%.1f" : "%+.1fx"), c[i], i);
00102     }
00103     printf("\n");
00104 }
00105 }
00106 return 0;
00107 }

```

## 1.2 main.c

```

00001 /** @file main.c
00002  * @brief 数式微分を行う
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 数式微分を行う
00012  *
00013  * 標準入力から n 次多項式の係数を読み込み、
00014  * その多項式と、微分した n-1 次多項式を表示する
00015  *
00016  * 入力：
00017  * - 標準入力の最初の整数は、多項式の次数 n (n は 0 以上の整数)
00018  * - それに引き続く n 個の実数は n 次多項式の係数
00019  *
00020  * 出力：
00021  * - n 次多項式を以下のフォーマットで出力
00022  *   c0+c1x+c2x^2+...+cnx^n
00023  * - 微分した n-1 次多項式を以下のフォーマットで出力
00024  *   c0+c1x+c2x^2+...+cnx^n-1
00025  * - 空白は表示しない
00026  * - 係数が負であれば、+記号を-にする
00027  * - 定数係数 (c0) が負であれば-記号を表示する。正であれば記号を表示しない
00028  * - 係数は小数点第 1 位まで表示する (%.1f) .
00029  *
00030  * 次数 n は、2 次以降の項は整数だけを表示し x^n とする。1 次の項は x と表示する。定数項は x を表示しない。
00031  *
00032  * 注意：
00033  * - 次数 n が負の場合には、何も表示せずに終了する (return 0 で)
00034  * - 次数 n が 0 の場合は定数項だけであり、微分したら 0 を表示する
00035  *
00036  * 入力例：
00037 \verbatim
00038 5 0.299015168361 -0.634498680199 0.076517409876 -0.155687714204 0.50453614886
00039 0.260700633183
00040 \endverbatim
00041 * 出力例：
00042 \verbatim
00043 0.3-0.6x+0.1x^2-0.2x^3+0.5x^4+0.3x^5
00044 -0.6+0.2x-0.5x^2+2.0x^3+1.3x^4
00045 \endverbatim
00046 * 入力例：
00047 \verbatim
00048 -1
00049 \endverbatim
00050 * 出力例：
00051 \verbatim
00052 \endverbatim
00053 * 入力例：

```

```

00054 \verbatim
00055 0 0.338757150218
00056 \endverbatim
00057 * 出力例:
00058 \verbatim
00059 0.3
00060 0.0
00061 \endverbatim
00062 * @date 2016/10/26
00063 * @author 佐伯雄飛, B162392
00064 */
00065 int main(void) {
00066     int n; // order of polynomial
00067     scanf("%d", &n);
00068     n++;
00069
00070     if (n < 1) {
00071         printf("");
00072     }
00073
00074     else {
00075         float c[n]; // coefficient of n-th order term
00076         for (int i = 0; i < n; i++) {
00077             scanf("%f", &c[i]);
00078         }
00079
00080         // print the polynomial: c0 + c1 x + c2 x^2 + .... + cn x^n
00081
00082         for (int i = 0; i < n; i++) {
00083             printf(i > 1 ? "%+.1fx^%d" : (i == 0 ? "%.1f" : "%+.1fx"), c[i], i);
00084         }
00085         printf("\n");
00086
00087         if (n == 1) {
00088             printf("0.0");
00089             printf("\n");
00090         }
00091
00092         else {
00093             // differentiation
00094             for (int i = 1; i < n; i++) {
00095                 c[i] *= i;
00096                 c[i - 1] = c[i];
00097             }
00098
00099             // print the derivative: c0' + c1' x + c2' x^2 + .... + cn' x^n-1
00100             for (int i = 0; i < n - 1; i++) {
00101                 printf(i > 1 ? "%+.1fx^%d" : (i == 0 ? "%.1f" : "%+.1fx"), c[i], i);
00102             }
00103             printf("\n");
00104         }
00105     }
00106     return 0;
00107 }

```