

# 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 12 月 15 日 (木) 15 時 07 分 32 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

文字列中の，大文字と小文字のヒストグラムを作成する

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

関数

- int `main` (int argc, char \*argv[])  
文字列中の，大文字と小文字のヒストグラムを作成する

#### 1.1.1 詳解

文字列中の，大文字と小文字のヒストグラムを作成する

日付

2016/12/15

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 ( int argc, char \* argv[ ] )

文字列中の，大文字と小文字のヒストグラムを作成する

入力：

- 標準入力には検索対象の，空白と改行を含む文字列が与えられる．長さは不明（ただし `max_int` 以下とする）．

出力：

- 文字列に含まれる大文字の数を，A からZ の順番に（`%5d` で）表示し，改行する．

- 文字列に含まれる小文字の数を、a から z の順番に (%5d で) 表示し、改行する。

入力例：

```
dpMhhyPmc yI DT iJUXpNfh RpV PPWJGvcE nuUB jqYjBrPqs
cS ysO rDE eRiZNQS YjwrGKI Ujg
OeurKNK zTSGVHej SxR iDmueNb
```

出力例 (フォーマットの都合上改行されています)：

```
      0      2      0      3      2      0      3      1      2      2      3      0      1
4      2      4      1      3      4      2      3      2      1      1      2      1
      0      1      3      1      4      1      1      3      3      5      0      0      2
1      0      3      2      4      2      0      3      1      1      1      3      1
```

入力例：

```
YjbiSz VdGaIJ
kJSJQNxNX YoPZY Rwe
FX zUUCazG lttNJ dVeNwI ZMEyXkOmg XGHW VBHJ
MkcR LdZT KHO LI InoJhFI
Yo JKSBedv CEgYo xvulXS pIIC wFj Lw
MRvqx mz gMMhJ
qfDxEEn DNX wdtGFdtBa uektgqxS EKDtEYMG pJq
Wdi XXLjM scXkn soxPTZs yNQggfT WTPeB
```

出力例 (フォーマットの都合上改行されています)：

```
      0      3      3      3      5      4      5      3      7      9      3      4      7
6      2      2      2      3      5      4      2      3      3      9      6      4
      3      2      2      7      5      2      6      2      2      3      5      2      2
3      5      3      4      0      3      6      2      3      5      6      2      4
```

日付

2016/12/15

著者

佐伯雄飛, B162392

[main.c](#) の行目に定義があります。

```
00057                                     {
00058     int* lower_letter_histogram;
00059     int* upper_letter_histogram;
00060
00061     lower_letter_histogram = (int*)malloc(26 * sizeof(int));
00062     upper_letter_histogram = (int*)malloc(26 * sizeof(int));
00063     if (lower_letter_histogram == NULL || upper_letter_histogram == NULL) {
00064         printf("can't alloc memory\n");
00065         exit(0);
00066     }
00067
00068     char c;
00069     while ((c = getchar()) != EOF) { // EOF is the end of stdin
00070         if (isupper(c)) {
00071             upper_letter_histogram[c - 'A']++;
00072             // 大文字ヒストグラム用のコードを書いてください
00073         }
00074         if (islower(c)) {
00075             lower_letter_histogram[c - 'a']++;
00076             // 小文字ヒストグラム用のコードを書いてください
00077         }
00078     }
00079
00080     for (int i = 0; i < 26; i++) {
00081         printf("%5d ", upper_letter_histogram[i]);
00082     }
00083     printf("\n");
00084     for (int i = 0; i < 26; i++) {
```

```

00085     printf("%5d ", lower_letter_histogram[i]);
00086 }
00087 printf("\n");
00088
00089 free(lower_letter_histogram);
00090 free(upper_letter_histogram);
00091
00092 return 0;
00093 }

```

## 1.2 main.c

```

00001 /** @file main.c
00002 * @brief 文字列中の、大文字と小文字のヒストグラムを作成する
00003 * @date 2016/12/15
00004 * @author 佐伯雄飛
00005 * @author B162392
00006 */
00007
00008 #include <stdio.h>
00009 #include <stdlib.h> // malloc
00010 #include <ctype.h> // isupper, islower
00011
00012 /** @fn int main(int argc, char* argv[])
00013 * @brief 文字列中の、大文字と小文字のヒストグラムを作成する
00014 *
00015 * 入力:
00016 * - 標準入力には検索対象の、空白と改行を含む文字列が与えられる。
00017 * 長さは不明（ただし max_int 以下とする）。
00018 *
00019 * 出力:
00020 * - 文字列に含まれる大文字の数を、A から Z の順番に（%5d で）表示し、改行する。
00021 * - 文字列に含まれる小文字の数を、a から z の順番に（%5d で）表示し、改行する。
00022 *
00023 * 入力例:
00024 \verbatim
00025 dpMhyPmc yI DT iJUXpNfh RpV PPWJGvcE nuUB jqYjBrPqs
00026 cS ysO rDE eRiZNQS YjwrGKI Ujg
00027 OeurKNK zTSGVHej SxR iDmueNb
00028 \endverbatim
00029 * 出力例（フォーマットの都合上改行されています）:
00030 \verbatim
00031 0 2 0 3 2 0 3 1 2 2 3 0 1
00032 4 2 4 1 3 4 2 3 2 1 1 2 1
00033 0 1 3 1 4 1 1 3 3 5 0 0 2
00034 1 0 3 2 4 2 0 3 1 1 1 3 1
00035 \endverbatim
00036 * 入力例:
00037 \verbatim
00038 YjbiSz VdGaIJ
00039 kJSJQNxNX YoPZY Rwe
00040 FX zUUCazG lttNJ dVeNwI ZMEyXkOmg XGHW VBHJ
00041 MkCR LdZT KHO LI InoJhFI
00042 Yo JKSBEdv CEgYo xvulXS pIIC wFj Lw
00043 MRvqx mz gMMhJ
00044 qfDxEn DNX wdtGFdtBa uektgqxS EKDtemyG pJq
00045 Wdi XXLjM scXkn soxPTZs yNQggfT WTPeB
00046 \endverbatim
00047 * 出力例（フォーマットの都合上改行されています）:
00048 \verbatim
00049 0 3 3 3 5 4 5 3 7 9 3 4 7
00050 6 2 2 2 3 5 4 2 3 3 9 6 4
00051 3 2 2 2 7 5 2 6 2 2 3 5 2 2
00052 3 5 3 4 0 3 6 2 3 5 6 2 4
00053 \endverbatim
00054 * @date 2016/12/15
00055 * @author 佐伯雄飛, B162392
00056 */
00057 int main(int argc, char* argv[]) {
00058     int* lower_letter_histogram;
00059     int* upper_letter_histogram;
00060
00061     lower_letter_histogram = (int*)malloc(26 * sizeof(int));
00062     upper_letter_histogram = (int*)malloc(26 * sizeof(int));
00063     if (lower_letter_histogram == NULL || upper_letter_histogram == NULL) {
00064         printf("can't alloc memory\n");
00065         exit(0);
00066     }
00067
00068     char c;
00069     while ((c = getchar()) != EOF) { // EOF is the end of stdin
00070         if (isupper(c)) {
00071             upper_letter_histogram[c - 'A']++;
00072             // 大文字ヒストグラム用のコードを書いてください

```

```
00073     }
00074     if (islower(c)) {
00075         lower_letter_histogram[c - 'a']++;
00076         // 小文字ヒストグラム用のコードを書いてください
00077     }
00078 }
00079
00080 for (int i = 0; i < 26; i++) {
00081     printf("%5d ", upper_letter_histogram[i]);
00082 }
00083 printf("\n");
00084 for (int i = 0; i < 26; i++) {
00085     printf("%5d ", lower_letter_histogram[i]);
00086 }
00087 printf("\n");
00088
00089 free(lower_letter_histogram);
00090 free(upper_letter_histogram);
00091
00092 return 0;
00093 }
```