# 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 11 月 29 日 (火) 21 時 44 分 21 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 . . . . . 5

# 1 ファイル詳解

## 1.1 main.c ファイル

メニュー項目を表示する

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

関数

- void show_items (char *items[], int n)
  **メニュー項目を表示する**
- void show_menu (char **menu[], int n)
  **メニューを表示する**
- int main (void)
  **メニュー項目を表示する**

### 1.1.1 詳解

メニュー項目を表示する

日付

2016/11/24

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 )

メニュー項目を表示する
入力：

- 標準入力から，メニュー項目を表すアルファベット 1 文字が与えられる

- もしくは終了 (quit) を意味する'q' が与えられる
- 文字が複数与えられる場合，空白（もしくは改行）で区切られている

出力：

- メニュー menu を表示する
- そして入力されたアルファベット 1 文字に対応するメニュー項目の内容を表示する．
- 文字'q' が与えられた場合，終了する．
- それ以外の文字が入力された場合，menu 表示に戻る
- 空白（もしくは改行）の後に続けて文字が与えられた場合，上記の処理を繰り返す．

注意：
- メニュー項目 `File` の場合，先頭の `1` 文字 `F` が対応するアルファベットである．
- 以下のコードでは `4` つのメニュー項目が定義されている．

```
char *file[] = {"File(F)", "open",        "close",    "new",      "quit"};
char *edit[] = {"Edit(E)", "undo",        "paste",    "copy",     "delete"};
char *view[] = {"View(V)", "fullscreen", "minimize", "large",    "small"};
char *help[] = {"Help(H)", "help",        "search",   "document", "website"};
char **menu[] = {file, edit, view, help};
```

これら以外にもメニューを追加することを考慮して，入力文字がF かE かを判定するというコードを書かず，一般的なコードを書くこと．
実行例（キーボード入力の場合）：

```
$ ./main
File(F) Edit(E) View(V) Help(H)
F
open close new quit
File(F) Edit(E) View(V) Help(H)
E
undo paste copy delete
File(F) Edit(E) View(V) Help(H)
V
fullscreen minimize large small
File(F) Edit(E) View(V) Help(H)
H
help search document website
File(F) Edit(E) View(V) Help(H)
D
File(F) Edit(E) View(V) Help(H)
q
$
```

標準入力例：

```
F F E V
```

出力例：

```
File(F) Edit(E) View(V) Help(H)
open close new quit
File(F) Edit(E) View(V) Help(H)
open close new quit
File(F) Edit(E) View(V) Help(H)
undo paste copy delete
File(F) Edit(E) View(V) Help(H)
fullscreen minimize large small
File(F) Edit(E) View(V) Help(H)
```

日付

2016/11/24

著者

佐伯雄飛，B162392

main.c の 行目に定義があります。

```
00110                {
00111   char* file[] = {"File(F)", "open", "close", "new", "quit"};
00112   char* edit[] = {"Edit(E)", "undo", "paste", "copy", "delete"};
00113   char* view[] = {"View(V)", "fullscreen", "minimize", "large", "small"};
00114   char* help[] = {"Help(H)", "help", "search", "document", "website"};
00115   char** menu[] = {file, edit, view, help};
00116   int num_menu = 4;
00117
00118   int len = 100;
00119   char input[100];
00120   bool is_end = false;
00121
00122   while (!is_end) {
00123     show_menu(menu, num_menu);
00124
00125     int return_value = scanf("%99s", input);
00126
00127     if (return_value == 0 ||    // if input didn't match to %s
00128         return_value == EOF) {  // if there are no input
00129       break;                    // break while
00130     }
00131
00132     if ('q' == input[0]) {  // if input is 'q'
00133       break;                // break while
00134     }
00135     // printf("%c",*(*menu[0]+5));
00136     for (int n = 0; n < num_menu; n++) {
00137       if (input[0] == *(*menu[n] + 5)) {
00138         show_items(menu[n], 4);
00139       }
00140
00141       // 入力文字がどのメニュー項目に相当するのかを判定し，show_items() で表示してください
00142     }
00143   }
00144
00145   return 0;
00146 }
```

#### 1.1.2.2 void show_items ( char * *items[]*, int *n* )

メニュー項目を表示する

引数

| | |
|---|---|
| *items* | メニュー項目 |
| *n* | 項目数 |

日付

2016/06/08

著者

Toru Tamaki

main.c の 行目に定義があります。

```
00018                                   {
00019   for (int i = 1; i <= n; i++) {
00020     printf("%s ", items[i]);
00021   }
00022   printf("\n");
00023 }
```

#### 1.1.2.3 show_menu ( char ** *menu[ ],* int *n* )

メニューを表示する

引数

| | |
|---|---|
| *menu* | メニュー |
| *n* | メニュー数 |

日付

2016/06/08

著者

Toru Tamaki

main.c の 行目に定義があります。

```
00032                                       {
00033   for (int i = 0; i < n; i++) {
00034     printf("%s ", menu[i][0]);
00035   }
00036   printf("\n");
00037 }
```

## 1.2 main.c

```
00001 /**  @file main.c
00002  *   @brief  メニュー項目を表示する
00003  *   @date   2016/11/24
00004  *   @author 佐伯雄飛
00005  *   @author B162392
00006  */
00007
00008 #include <stdio.h>
00009 #include <stdbool.h>  // bool
00010
00011 /** @fn     void show_items(char *items[], int n)
00012  *  @brief  メニュー項目を表示する
00013  *  @param  items メニュー項目
00014  *  @param  n 項目数
00015  *  @date   2016/06/08
00016  *  @author Toru Tamaki
00017  */
00018 void show_items(char* items[], int n) {
00019   for (int i = 1; i <= n; i++) {
00020     printf("%s ", items[i]);
00021   }
00022   printf("\n");
00023 }
00024
00025 /** @fn     show_menu(char **menu[], int n)
00026  *  @brief  メニューを表示する
00027  *  @param  menu メニュー
00028  *  @param  n メニュー数
00029  *  @date   2016/06/08
00030  *  @author Toru Tamaki
00031  */
00032 void show_menu(char** menu[], int n) {
00033   for (int i = 0; i < n; i++) {
00034     printf("%s ", menu[i][0]);
00035   }
00036   printf("\n");
00037 }
00038
00039 /** @fn int main(void)
00040  *  @brief メニュー項目を表示する
00041  *
00042  *  入力：
00043  *  - 標準入力から，メニュー項目を表すアルファベット 1 文字が与えられる
00044  *  - もしくは終了 (quit) を意味する'q' が与えられる
00045  *  - 文字が複数与えられる場合，空白（もしくは改行）で区切られている
00046  *
00047  *  出力：
00048  *  - メニュー menu を表示する
00049  *  -
00050 そして入力されたアルファベット 1 文字に対応するメニュー項目の内容を表示する.
00051  *  - 文字'q' が与えられた場合，終了する.
00052  *  - それ以外の文字が入力された場合，menu 表示に戻る
```

```
00053  *  -
00054 空白（もしくは改行）の後に続けて文字が与えられた場合，上記の処理を繰り返す.
00055  *
00056  *  注意：
00057  *  - メニュー項目 File の場合，先頭の 1 文字 F が対応するアルファベットである.
00058  *  - 以下のコードでは 4 つのメニュー項目が定義されている.
00059 \verbatim
00060   char *file[] = {"File(F)", "open",       "close",    "new",      "quit"};
00061   char *edit[] = {"Edit(E)", "undo",       "paste",    "copy",     "delete"};
00062   char *view[] = {"View(V)", "fullscreen", "minimize", "large",    "small"};
00063   char *help[] = {"Help(H)", "help",       "search",   "document", "website"};
00064   char **menu[] = {file, edit, view, help};
00065 \endverbatim
00066  *
00067 これら以外にもメニューを追加することを考慮して，入力文字が F か E かを判定するというコードを書かず，
00068  *     一般的なコードを書くこと.
00069  *
00070  *  実行例（キーボード入力の場合）：
00071 \verbatim
00072 $ ./main
00073 File(F) Edit(E) View(V) Help(H)
00074 F
00075 open close new quit
00076 File(F) Edit(E) View(V) Help(H)
00077 E
00078 undo paste copy delete
00079 File(F) Edit(E) View(V) Help(H)
00080 V
00081 fullscreen minimize large small
00082 File(F) Edit(E) View(V) Help(H)
00083 H
00084 help search document website
00085 File(F) Edit(E) View(V) Help(H)
00086 D
00087 File(F) Edit(E) View(V) Help(H)
00088 q
00089 $
00090 \endverbatim
00091   *  標準入力例：
00092 \verbatim
00093 F F E V
00094 \endverbatim
00095   *  出力例：
00096 \verbatim
00097 File(F) Edit(E) View(V) Help(H)
00098 open close new quit
00099 File(F) Edit(E) View(V) Help(H)
00100 open close new quit
00101 File(F) Edit(E) View(V) Help(H)
00102 undo paste copy delete
00103 File(F) Edit(E) View(V) Help(H)
00104 fullscreen minimize large small
00105 File(F) Edit(E) View(V) Help(H)
00106 \endverbatim
00107  *  @date   2016/11/24
00108  *  @author 佐伯雄飛, B162392
00109  */
00110 int main(void) {
00111   char* file[] = {"File(F)", "open", "close", "new", "quit"};
00112   char* edit[] = {"Edit(E)", "undo", "paste", "copy", "delete"};
00113   char* view[] = {"View(V)", "fullscreen", "minimize", "large", "small"};
00114   char* help[] = {"Help(H)", "help", "search", "document", "website"};
00115   char** menu[] = {file, edit, view, help};
00116   int num_menu = 4;
00117
00118   int len = 100;
00119   char input[100];
00120   bool is_end = false;
00121
00122   while (!is_end) {
00123     show_menu(menu, num_menu);
00124
00125     int return_value = scanf("%99s", input);
00126
00127     if (return_value == 0 ||    // if input didn't match to %s
00128         return_value == EOF) {  // if there are no input
00129       break;                    // break while
00130     }
00131
00132     if ('q' == input[0]) {  // if input is 'q'
00133       break;                // break while
00134     }
00135     // printf("%c",*(*menu[0]+5));
00136     for (int n = 0; n < num_menu; n++) {
00137       if (input[0] == *(*menu[n] + 5)) {
00138         show_items(menu[n], 4);
00139       }
```

```
00140
00141       // 入力文字がどのメニュー項目に相当するのかを判定し，show_items() で表示してください
00142     }
00143   }
00144
00145   return 0;
00146 }
```