

# 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

2017 年 01 月 10 日 (火) 13 時 20 分 40 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

課程配属条件の判定

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

関数

- int `main` (int argc, char \*argv[])  
課程配属条件の判定

#### 1.1.1 詳解

課程配属条件の判定

日付

2017/1/10

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[] )

課程配属条件の判定

入力：

- 標準入力には、まず学生数が与えられる
- 続いて講義数が与えられる。それに引き続いて、空白で区切られた講義コード（6桁）が講義数分与えられる。
- 続いて「学生番号（6桁）、講義コード、素点（100点満点）」が（空白で区切られて）与えられる。このフォーマットは、一人の学生について異なる講義コードが連続しており、その学生の成績がすべて与えられたら、別の学生の成績が与えられる。

- ・ 仮定：すべての学生は同じ講義を受講していると仮定する（入力はこの仮定を満たしている）

出力：

- ・ 各学生について、課程配属条件を満たしているかどうかを表示する。
- ・ ある学生が 32 単位以上を取得している場合、課程配属条件満たす。
- ・ ある講義の点数が 60 点以上のとき、その講義に合格し、2 単位を取得する。60 点未満であれば不合格であり、単位を取得できない。
- ・ 出力は、学生番号（%06d）、取得単位数（d）、判定結果（"fail"か"pass"）の順に空白で区切り、改行する。

入力例（途中省略）:

```
150
20
030323 044417 112271 166084 230334 288383 304507 343261 375762 408164 415311
417011 467643 486000 518187 618065 674215 712181 755421 883186
002487 030323 74
002487 044417 41
002487 112271 64
002487 166084 38
002487 230334 72
002487 288383 82
002487 304507 67
...
886870 417011 85
886870 467643 61
886870 486000 75
886870 518187 88
886870 618065 76
886870 674215 73
886870 712181 69
886870 755421 90
886870 883186 83
```

出力例（途中省略）:

```
002487 34 pass
023065 32 pass
038034 30 fail
038177 32 pass
044436 36 pass
050530 34 pass
...
874131 40 pass
875426 34 pass
880624 34 pass
881667 38 pass
882464 38 pass
886516 30 fail
886870 36 pass
```

日付

2017/1/10

著者

佐伯雄飛，B162392

[main.c](#) の行目に定義があります。

```

00076             {
00077     int num_students;
00078     scanf("%d", &num_students);
00079
00080     int num_class;
00081     scanf("%d", &num_class);
00082
00083     int class_codes[num_class];
00084     for (int i = 0; i < num_class; i++) {
00085         scanf("%d", &class_codes[i]);
00086     }
00087
00088     int student_id, class_code, score;
00089
00090     for (int s = 0; s < num_students; s++) {
00091         int a = 0;
00092         for (int c = 0; c < num_class; c++) {
00093             scanf("%d", &student_id);
00094             scanf("%d", &class_code);
00095             scanf("%d", &score);
00096             if (score >= 60) {
00097                 a += 2;
00098             }
00099         }
00100         if (a > 31)
00101             printf("%06d %d pass\n", student_id, a);
00102         else
00103             printf("%06d %d fail\n", student_id, a);
00104     }
00105
00106     return 0;
00107 }

```

## 1.2 main.c

```

00001 /** @file main.c
00002  * @brief 課程配属条件の判定
00003  * @date 2017/1/10
00004  * @author 佐伯雄飛
00005  * @author B162392
00006 */
00007
00008 #include <stdio.h>
00009
00010 /** @fn int main(int argc, char* argv[])
00011  * @brief 課程配属条件の判定
00012  *
00013  * 入力:
00014  * - 標準入力には、まず学生数が与えられる
00015  * - 続いて講義数が与えられる。それに引き続いて、
00016  *   空白で区切られた講義コード(6桁)が講義数分与えられる。
00017  * - 続いて「学生番号(6桁)、講義コード、素点(100点満点)」が
00018  *   (空白で区切られて)与えられる。
00019  *   このフォーマットは、一人の学生について異なる講義コードが連続しており、
00020  *   その学生の成績がすべて与えられたら、別の学生の成績が与えられる。
00021  * - 仮定: すべての学生は同じ講義を受講していると仮定する
00022  *   (入力はこの仮定を満たしている)
00023  *
00024  * 出力:
00025  * - 各学生について、課程配属条件を満たしているかどうかを表示する。
00026  * - ある学生が 32 単位以上を取得している場合、課程配属条件満たす。
00027  * - ある講義の点数が 60 点以上のとき、その講義に合格し、2 単位を取得する。
00028  *   60 点未満であれば不合格であり、単位を取得できない。
00029  * - 出力は、学生番号(%06d)、取得単位数(%d)、
00030  *   判定結果("fail"か"pass")の順に空白で区切り、改行する。
00031  *
00032  * 入力例(途中省略):
00033 \verbatim
00034 150
00035 20
00036 030323 044417 112271 166084 230334 288383 304507 343261 375762 408164 415311
00037 417011 467643 486000 518187 618065 674215 712181 755421 883186
00038 002487 030323 74
00039 002487 044417 41
00040 002487 112271 64
00041 002487 166084 38
00042 002487 230334 72
00043 002487 288383 82
00044 002487 304507 67
00045 ...
00046 886870 417011 85
00047 886870 467643 61
00048 886870 486000 75
00049 886870 518187 88

```

```
00050 886870 618065 76
00051 886870 674215 73
00052 886870 712181 69
00053 886870 755421 90
00054 886870 883186 83
00055 \endverbatim
00056 * 出力例 (途中省略):
00057 \verbatim
00058 002487 34 pass
00059 023065 32 pass
00060 038034 30 fail
00061 038177 32 pass
00062 044436 36 pass
00063 050530 34 pass
00064 ...
00065 874131 40 pass
00066 875426 34 pass
00067 880624 34 pass
00068 881667 38 pass
00069 882464 38 pass
00070 886516 30 fail
00071 886870 36 pass
00072 \endverbatim
00073 * @date 2017/1/10
00074 * @author 佐伯雄飛, B162392
00075 */
00076 int main(int argc, char* argv[]) {
00077     int num_students;
00078     scanf("%d", &num_students);
00079
00080     int num_class;
00081     scanf("%d", &num_class);
00082
00083     int class_codes[num_class];
00084     for (int i = 0; i < num_class; i++) {
00085         scanf("%d", &class_codes[i]);
00086     }
00087
00088     int student_id, class_code, score;
00089
00090     for (int s = 0; s < num_students; s++) {
00091         int a = 0;
00092         for (int c = 0; c < num_class; c++) {
00093             scanf("%d", &student_id);
00094             scanf("%d", &class_code);
00095             scanf("%d", &score);
00096             if (score >= 60) {
00097                 a += 2;
00098             }
00099         }
00100         if (a > 31)
00101             printf("%06d %d pass\n", student_id, a);
00102         else
00103             printf("%06d %d fail\n", student_id, a);
00104     }
00105
00106     return 0;
00107 }
```