

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 37 分 10 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

GPA の計算

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

関数

- int `main` (int argc, char *argv[])
GPA の計算

1.1.1 詳解

GPA の計算

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[])

GPA の計算

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，GPA を計算する．
- ・ ある講義の成績：点数が 90 点～100 点とき，秀．点数が 80 点～89 点とき，優．点数が 70 点～79 点とき，良．点数が 60 点～69 点とき，可．
- ・ 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．
- ・ $GPA = (\text{秀の単位数} * 4 + \text{優の単位数} * 3 + \text{良の単位数} * 2 + \text{可の単位数} * 1) / (\text{総登録単位数} * 4)$

– 100

- ・ 出力は，学生番号（%06d），GPA（%.2f）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 48.75
023065 48.75
038034 42.50
038177 45.00
044436 57.50
050530 46.25
...
874131 62.50
875426 47.50
880624 43.75
881667 50.00
882464 57.50
886516 38.75
886870 56.25
```

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。

```
00080                                     {
00081     int num_students;
00082     scanf("%d", &num_students);
00083
00084     int num_class;
00085     scanf("%d", &num_class);
00086
00087     int class_codes[num_class];
00088     for (int i = 0; i < num_class; i++) {
00089         scanf("%d", &class_codes[i]);
00090     }
00091
00092     int student_id, class_code, score;
00093
00094     for (int s = 0; s < num_students; s++) {
00095         float a = 0;
00096         float b = 0;
00097         for (int c = 0; c < num_class; c++) {
00098             scanf("%d", &student_id);
00099             scanf("%d", &class_code);
00100             scanf("%d", &score);
00101             if (score >= 90)
00102                 a += 4;
00103             else if (score >= 80)
00104                 a += 3;
00105             else if (score >= 70)
00106                 a += 2;
00107             else if (score >= 60)
00108                 a += 1;
00109         }
00110         int d = num_class * 4;
00111         b = a / d * 100;
00112
00113         printf("%06d %.2f\n", student_id, b);
00114     }
00115
00116     return 0;
00117 }
```

1.2 main.c

```
00001 /** @file main.c
00002  * @brief GPA の計算
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 GPA の計算
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、GPA を計算する。
00026  * - ある講義の成績: 点数が 90 点 ~ 100 点とき、秀。
00027  *   点数が 80 点 ~ 89 点とき、優。
00028  *   点数が 70 点 ~ 79 点とき、良。
00029  *   点数が 60 点 ~ 69 点とき、可。
00030  * - 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。
00031  * - GPA = (秀の単位数 * 4 + 優の単位数 * 3 + 良の単位数 * 2 + 可の単位数 * 1)
00032  *   / (総登録単位数 * 4)
00033  *   * 100
00034  * - 出力は、学生番号 (%06d), GPA (%.2f) の順に、空白で区切り、改行する。
00035  *
```

```

00036  * 入力例 (途中省略):
00037  \verbatim
00038  150
00039  20
00040  030323 044417 112271 166084 230334 288383 304507 343261 375762 408164 415311
00041  417011 467643 486000 518187 618065 674215 712181 755421 883186
00042  002487 030323 74
00043  002487 044417 41
00044  002487 112271 64
00045  002487 166084 38
00046  002487 230334 72
00047  002487 288383 82
00048  002487 304507 67
00049  ...
00050  886870 417011 85
00051  886870 467643 61
00052  886870 486000 75
00053  886870 518187 88
00054  886870 618065 76
00055  886870 674215 73
00056  886870 712181 69
00057  886870 755421 90
00058  886870 883186 83
00059  \endverbatim
00060  * 出力例 (途中省略):
00061  \verbatim
00062  002487 48.75
00063  023065 48.75
00064  038034 42.50
00065  038177 45.00
00066  044436 57.50
00067  050530 46.25
00068  ...
00069  874131 62.50
00070  875426 47.50
00071  880624 43.75
00072  881667 50.00
00073  882464 57.50
00074  886516 38.75
00075  886870 56.25
00076  \endverbatim
00077  * @date 2017/1/10
00078  * @author 佐伯雄飛, B162392
00079  */
00080  int main(int argc, char* argv[]) {
00081      int num_students;
00082      scanf("%d", &num_students);
00083
00084      int num_class;
00085      scanf("%d", &num_class);
00086
00087      int class_codes[num_class];
00088      for (int i = 0; i < num_class; i++) {
00089          scanf("%d", &class_codes[i]);
00090      }
00091
00092      int student_id, class_code, score;
00093
00094      for (int s = 0; s < num_students; s++) {
00095          float a = 0;
00096          float b = 0;
00097          for (int c = 0; c < num_class; c++) {
00098              scanf("%d", &student_id);
00099              scanf("%d", &class_code);
00100              scanf("%d", &score);
00101              if (score >= 90)
00102                  a += 4;
00103              else if (score >= 80)
00104                  a += 3;
00105              else if (score >= 70)
00106                  a += 2;
00107              else if (score >= 60)
00108                  a += 1;
00109          }
00110          int d = num_class * 4;
00111          b = a / d * 100;
00112
00113          printf("%06d %.2f\n", student_id, b);
00114      }
00115
00116      return 0;
00117 }

```