lab8.c U | output.txt U | Makefile U

Lab8 > output.txt

```
1  Dodson Steven A0858775 96 92
2  Lewis Stacey A0284552 99 97
3  Morrow Lauren A0916826 90 90
4  Perry Eric A0730019 97 98
5  Perry Eric A0730019 98 97
6  Perry Eric A0730020 98 97
7  Webb Shawn A0526252 87 94
8
```

ref1.txt U

Lab8 > ref1.txt

```
1  Dodson Steven A0858775 96 92
2  Lewis Stacey A0284552 99 97
3  Morrow Lauren A0916826 90 90
4  Perry Eric A0730019 97 98
5  Perry Eric A0730019 98 97
6  Perry Eric A0730020 98 97
7  Webb Shawn A0526252 87 94
8
```

OUTPUT DEBUG CONSOLE TERMINAL PORTS zsh - lab8

```
mskim ~/workspace/comp2510/lab8 main make
gcc -Wall -std=c11 lab8.c -o lab8.out
mskim ~/workspace/comp2510/lab8 main make convert_input
mskim ~/workspace/comp2510/lab8 main make run
./lab8.out input1.txt output.txt
mskim ~/workspace/comp2510/lab8 main make clean
rm -f lab8.out
mskim ~/workspace/comp2510/lab8 main make convert_output
mskim ~/workspace/comp2510/lab8 main make check
Pass
mskim ~/workspace/comp2510/lab8 main
```

lab8.c U | output.txt U | Makefile U | ref2.txt U | ref3.txt U | ref4.txt U | ref5.txt

Lab8 > output.txt

```
1  Gilbert Howard A0212656 89 73
2  Griffin Benjamin A0699185 64 96
3  Lara Robert A0368288 80 80
4  Lara Robert A0368288 81 80
5  Lara Robert A0368289 80 81
6  Lee Patrick A0893874 95 73
7  Montgomery Jamie A0962876 87 74
8
```

Lab8 > ref2.txt

```
1  Gilbert Howard A0212656 89 73
2  Griffin Benjamin A0699185 64 96
3  Lara Robert A0368288 80 80
4  Lara Robert A0368288 81 80
5  Lara Robert A0368289 80 81
6  Lee Patrick A0893874 95 73
7  Montgomery Jamie A0962876 87 74
8
```

OUTPUT DEBUG CONSOLE TERMINAL PORTS zsh - lab8

```
mskim ~/workspace/comp2510/lab8 main make
gcc -Wall -std=c11 lab8.c -o lab8.out
mskim ~/workspace/comp2510/lab8 main make convert_input
mskim ~/workspace/comp2510/lab8 main make run
./lab8.out input2.txt output.txt
mskim ~/workspace/comp2510/lab8 main make clean
rm -f lab8.out
mskim ~/workspace/comp2510/lab8 main make convert_output
mskim ~/workspace/comp2510/lab8 main make check
Pass
mskim ~/workspace/comp2510/lab8 main
```

lab8.c U | output.txt U | Makefile U

Lab8 > output.txt

```
Anderson Christina A0973331 56 100
Bailey Aaron A0208249 52 99
Berry Jacqueline A0483351 87 68
Campbell Matthew A0622975 76 83
Fowler Clayton A0513357 70 70
Johnson Kyle A0273834 57 88
Johnston Maria A0413808 97 49
Malone Jacob A0962010 57 87
Morales Katherine A0253258 93 49
Reed Robert A0657526 74 78
Richards Shawn A0942153 62 91
Richards Veronica A0958309 49 93
Rios Stacey A0296736 69 83
Williams Richard A0971344 96 46
```

U | ref2.txt U | ref3.txt U | ref4.txt U | ref5.txt

Lab8 > ref3.txt

```
Anderson Christina A0973331 56 100
Bailey Aaron A0208249 52 99
Berry Jacqueline A0483351 87 68
Campbell Matthew A0622975 76 83
Fowler Clayton A0513357 70 70
Johnson Kyle A0273834 57 88
Johnston Maria A0413808 97 49
Malone Jacob A0962010 57 87
Morales Katherine A0253258 93 49
Reed Robert A0657526 74 78
Richards Shawn A0942153 62 91
Richards Veronica A0958309 49 93
Rios Stacey A0296736 69 83
Williams Richard A0971344 96 46
```

OUTPUT DEBUG CONSOLE TERMINAL PORTS

zsh - lab8

```
mskim ~/workspace/comp2510/lab8 main make
gcc -Wall -std=c11 lab8.c -o lab8.out
mskim ~/workspace/comp2510/lab8 main make convert_input
mskim ~/workspace/comp2510/lab8 main make run
./lab8.out input3.txt output.txt
mskim ~/workspace/comp2510/lab8 main make clean
rm -f lab8.out
mskim ~/workspace/comp2510/lab8 main make convert_output
mskim ~/workspace/comp2510/lab8 main make check
Pass
mskim ~/workspace/comp2510/lab8 main
```

lab8.c U | output.txt U | Makefile U | ref2.txt U | ref3.txt U | ref4.txt U | ref5.t

Lab8 > output.txt

```
1  Bailey Candice A0834662 98 38
2  Brown Shane A0738510 57 70
3  Brown Tracy A0819034 95 33
4  Callahan Amy A0435370 68 67
5  Castillo Shawn A0908873 65 58
6  Gray Teresa A0797916 60 60
7  Hicks David A0858335 57 63
8  May Ryan A0760904 39 97
9  Miranda David A0330182 86 48
10 Patel Derrick A0570531 64 70
11 Patterson Sandra A0848894 41 94
12 Porter Michael A0922166 64 69
13 Terry Marcia A0996692 43 77
14 Washington Kenneth A0696548 51 77
15 Watson Mitchell A0735949 66 54
16
```

Lab8 > ref4.txt

```
1  Bailey Candice A0834662 98 38
2  Brown Shane A0738510 57 70
3  Brown Tracy A0819034 95 33
4  Callahan Amy A0435370 68 67
5  Castillo Shawn A0908873 65 58
6  Gray Teresa A0797916 60 60
7  Hicks David A0858335 57 63
8  May Ryan A0760904 39 97
9  Miranda David A0330182 86 48
10 Patel Derrick A0570531 64 70
11 Patterson Sandra A0848894 41 94
12 Porter Michael A0922166 64 69
13 Terry Marcia A0996692 43 77
14 Washington Kenneth A0696548 51 77
15 Watson Mitchell A0735949 66 54
16
```

OUTPUT DEBUG CONSOLE TERMINAL PORTS

zsh - lab8

```
mskim ~/workspace/comp2510/lab8 main make
gcc -Wall -std=c11 lab8.c -o lab8.out
mskim ~/workspace/comp2510/lab8 main make convert_input
mskim ~/workspace/comp2510/lab8 main make run
./lab8.out input4.txt output.txt
mskim ~/workspace/comp2510/lab8 main make clean
rm -f lab8.out
mskim ~/workspace/comp2510/lab8 main make convert_output
mskim ~/workspace/comp2510/lab8 main make check
Pass
mskim ~/workspace/comp2510/lab8 main
```

lab8.c U | output.txt U | Makefile U | ref2.txt U | ref3.txt U | ref4.txt U | ref5.txt

Lab8 > output.txt

```
41  Pratt Ricky A0768873 58 39
42  Rivas Jeremy A0305368 89 12
43  Scott Cindy A0888243 49 30
44  Simon Thomas A0517482 50 56
45  Singleton Jordan A0566335 59 10
46  Small Alexander A0127412 70 13
47  Smith Adam A0782612 81 0
48  Smith Anthony A0410478 41 28
49  Stewart Alison A0772659 31 39
50  Stone James A0364579 79 19
51  Summers Michael A0893440 55 50
52  Swanson Alan A0730535 33 18
53  Taylor Andrew A0139853 49 13
54  Valdez Abigail A0760928 23 30
55  Vang Jonathan A0696603 59 33
56  Wagner Kylie A0950922 38 58
57  Ward Alexis A0165533 48 32
58  Ward Mark A0460420 26 44
59  Wells Brittany A0660122 13 6
60  Wilkinson Robert A0133880 58 34
61  Williams Kimberly A0731536 1 6
62
```

Lab8 > ref5.txt

```
41  Pratt Ricky A0768873 58 39
42  Rivas Jeremy A0305368 89 12
43  Scott Cindy A0888243 49 30
44  Simon Thomas A0517482 50 56
45  Singleton Jordan A0566335 59 10
46  Small Alexander A0127412 70 13
47  Smith Adam A0782612 81 0
48  Smith Anthony A0410478 41 28
49  Stewart Alison A0772659 31 39
50  Stone James A0364579 79 19
51  Summers Michael A0893440 55 50
52  Swanson Alan A0730535 33 18
53  Taylor Andrew A0139853 49 13
54  Valdez Abigail A0760928 23 30
55  Vang Jonathan A0696603 59 33
56  Wagner Kylie A0950922 38 58
57  Ward Alexis A0165533 48 32
58  Ward Mark A0460420 26 44
59  Wells Brittany A0660122 13 6
60  Wilkinson Robert A0133880 58 34
61  Williams Kimberly A0731536 1 6
62
```

OUTPUT DEBUG CONSOLE TERMINAL PORTS

zsh - lab8

```
mskim ~/workspace/comp2510/lab8 main make
gcc -Wall -std=c11 lab8.c -o lab8.out
mskim ~/workspace/comp2510/lab8 main make convert_input
mskim ~/workspace/comp2510/lab8 main make run
./lab8.out input5.txt output.txt
mskim ~/workspace/comp2510/lab8 main make clean
rm -f lab8.out
mskim ~/workspace/comp2510/lab8 main make convert_output
mskim ~/workspace/comp2510/lab8 main make check
Pass
mskim ~/workspace/comp2510/lab8 main
```

C lab8.c U ×

Lab8 > C lab8.c > swap(Student *, Student *)

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct {
    char lName[50];
    char fName[50];
    char studentNum[20];
    int  midGrade;
    int  finalGrade;
} Student;

void swap(Student *a, Student *b)
{
    Student tmp = *a;
    *a = *b;
    *b = tmp;
}

int comesBefore(const Student *x, const Student *y)
{
    int r = strcmp(x->lName, y->lName);
    if (r) return r < 0;
    r = strcmp(x->fName, y->fName);
    if (r) return r < 0;
    r = strcmp(x->studentNum, y->studentNum);
    if (r) return r < 0;
    if (x->midGrade != y->midGrade)   return x->midGrade < y->midGrade;
    if (x->finalGrade != y->finalGrade) return x->finalGrade < y->finalGrade;

    return 0;
}

void selectionSort(Student arr[], int n)
{
    for (int i = 0; i < n - 1; i++) {
        int min = i;
        for (int j = i + 1; j < n; j++) {
            if (comesBefore(&arr[j], &arr[min])) {
                min = j;
            }
        }
        if (min != i) swap(&arr[min], &arr[i]);
    }
```

C lab8.c U ×

Lab8 > C lab8.c > swap(Student *, Student *)

```
34  void selectionSort(Student arr[], int n)
45  }
46
47  int main(int argc, char **argv){
48
49      if (argc != 3)
50      {
51          return 1;
52      }
53
54      FILE *in = fopen(argv[1], "r");
55      if (in == NULL)
56      {
57          return 1;
58      }
59
60      FILE *out = fopen(argv[2], "w");
61      if (out == NULL)
62      {
63          fclose(in);
64          return 1;
65      }
66
67      int option;
68      if (fscanf(in, "%d", &option) != 1 || option < 1 || option > 5)
69      {
70          fprintf(out, "Error");
71          fclose(in);
72          fclose(out);
73          return 0;
74      }
75
76      int count = 0;
77      int c;
78
79      while ((c = fgetc(in)) != EOF) {
80          if (c == '\n') {
81              count++;
82          }
83      }
84
85      rewind(in);
86
87      // code to put student list
```

C lab8.c U ×

Lab8 > C lab8.c > swap(Student *, Student *)

```
88      if (fscanf(in, "%d", &option) != 1 || option < 1 || option > 5)
89      {
90          fprintf(out, "Error");
91          fclose(in);
92          fclose(out);
93
94          return 0;
95      }
96
97      Student *students = malloc(count * sizeof(Student));
98      if (count > 0 && students == NULL)
99      {
100         fprintf(out, "Error");
101         fclose(in);
102         fclose(out);
103
104         return 0;
105     }
106
107     int n = 0;
108     char last[50];
109
110     while(fscanf(in, "%49s", last) == 1 && strcmp(last, "E") != 0)
111     {
112         if (n >= count)
113         {
114             fprintf(out, "Error");
115             free(students);
116             fclose(in);
117             fclose(out);
118
119             return 0;
120         }
121
122         if (fscanf(in, "%49s %19s %d %d",
123                 students[n].fName,
124                 students[n].studentNum,
125                 &students[n].midGrade,
126                 &students[n].finalGrade) != 4)
127         {
128             fprintf(out, "Error");
129             free(students);
130             fclose(in);
131             fclose(out);
```

C lab8.c U ×

Lab8 > C lab8.c > swap(Student *, Student *)

```
131             fclose(out);
132
133             return 0;
134         }
135
136         strcpy(students[n].lName, last);
137
138         if (students[n].lName[0] == '\0' || students[n].fName[0] == '\0')
139         {
140             fprintf(out, "Error");
141             free(students);
142             fclose(in);
143             fclose(out);
144
145             return 0;
146         }
147
148         char *id = students[n].studentNum;
149         int ok = (strlen(id) == 8 && id[0] == 'A');
150
151         for (int i = 1; ok && i < 8; i++)
152         {
153             if (id[i] < '0' || id[i] > '9') ok = 0;
154         }
155         if (!ok)
156         {
157             fprintf(out, "Error");
158             free(students);
159             fclose(in);
160             fclose(out);
161
162             return 0;
163         }
164
165         if (students[n].midGrade < 0 ||
166             students[n].midGrade > 100 ||
167             students[n].finalGrade < 0 ||
168             students[n].finalGrade > 100)
169         {
170             fprintf(out, "Error");
171             free(students);
172             fclose(in);
173             fclose(out);
174
```

C lab8.c U ×

Lab8 > C lab8.c > swap(Student *, Student *)

```c
            fclose(out);

            return 0;
        }

        n++;
    }

    Student *filtered = NULL;
    int fcount = 0;

    if (n > 0)
    {
        filtered = malloc(n * sizeof(Student));
        if (!filtered)
        {
            fprintf(out, "Error");
            free(students);
            fclose(in);
            fclose(out);

            return 0;
        }
    }

    for (int i = 0; i < n; i++)
    {
        double avg = (students[i].midGrade + students[i].finalGrade) / 2.0;

        int keep = 0;
        switch (option) {
            case 1: keep = (avg >= 90.0); break;
            case 2: keep = (avg >= 80.0 && avg < 90.0); break;
            case 3: keep = (avg >= 70.0 && avg < 80.0); break;
            case 4: keep = (avg >= 60.0 && avg < 70.0); break;
            case 5: keep = (avg < 60.0); break;
        }

        if (keep)
        {
            filtered[fcount++] = students[i];
        }
    }

```

C lab8.c U ×

Lab8 > C lab8.c > swap(Student *, Student *)

```
213                 filtered[fcount++] = students[i];
214             }
215         }
216
217         free(students);
218         students = filtered;
219         n = fcount;
220
221         selectionSort(students, n);
222
223         for (int i = 0; i < n; i++)
224         {
225             fprintf(out, "%s %s %s %d %d\n",
226                     students[i].lName,
227                     students[i].fName,
228                     students[i].studentNum,
229                     students[i].midGrade,
230                     students[i].finalGrade);
231         }
232
233         fclose(in);
234         fclose(out);
235         free(students);
236
237         return 0;
238     }
```