lab11

$ gcc lab11.c

$ ./a.out < lab11.dat

Grand Prize:

1: Ava BROWN 285.2

2: John PRICE 284.9

3: Abigail WILSON 277.3

4: Isaac WASHINGTON 276.4

5: Leah YOUNG 267.3

6: Samuel BENNETT 264.5

7: Caleb HENDERSON 263.2

8: Alexis JACKSON 261.8

Math Prize:

1: Gabriella HILL 99.3

2: Elizabeth ANDERSON 98.9

3: Benjamin RAMIREZ 97.4

4: Isaiah BUTLER 97.4

5: Daniel MORGAN 97.1

6: Alexa PEREZ 96.8

7: Alexander ROGERS 95.8

8: Anna HERNANDEZ 95.4

9: Christian BROOKS 95.3

10: Matthew RICHARDSON 94.1

Science Prize:

1: James PETERSON 98.7

2: Isaiah BUTLER 98.3

3: Carter HAYES 96.6

4: Dylan BARNES 96.3

5: Noah MURPHY 95.5

6: Avery LOPEZ 94

7: Sydney EVANS 93.1

8: Nicholas COLEMAN 91.7

9: Nevaeh SCOTT 88.5

10: Michael MORRIS 85.5

Literature Prize:

1: Elijah JAMES 99.7

2: Jack SIMMONS 99.6

3: Michael MORRIS 99.4

4: Natalie MARTIN 98.9

5: Nevaeh SCOTT 98.4

1

6: Alyssa MARTINEZ 97.8

7: James PETERSON 95.9

8: Logan TORRES 95.7

9: Audrey EDWARDS 95.5

10: Alexa PEREZ 93.2

score: 86.0

o. [Output] Program output is correct, good. o. [Format] Program format can be improved. o. [Header] comments need to be complete. o. [main] function needs a return statement. o. [Local] array 'score' is not needed.

2

lab11.c

1 // EE231002 Lab11. Academic Competition

2 // 111060023, Berlin

Need your Chinese name.

3 // Date: 2022/12/5

4

5 #include <stdio.h>

6 #include <stdlib.h>

7 #include <string.h>

8

9 struct STU {

10 char fName[15]; // structure definition

11 char lName[15]; // last name

12 double math, sci, lit; // test scores

13 double min; // minimum subject score

14 };

15 struct STU list[100]; // student list

16

17 void GrandPrize(struct STU list[100]); // print grand prize winner 18 void printScore(double score[100], int subPrize); // print score by case 19

20 int main(void)

21 {

22 int i, j; // loop control 23 double score[100]; // array to store score 24

25 scanf("FirstName LastName Math Science Literature\n"); // read in title 26 for (i = 0; i < 100; i++) { // read 100 student 27 scanf("%s %s %lf %lf %lf\n",

28 list[i].fName, list[i].lName, // students' names 29 &list[i].math, &list[i].sci, &list[i].lit); // students' scores 30 list[i].min = list[i].math; // assume math is the min 31 if (list[i].min > list[i].sci) { // if sci. is lower 32 list[i].min = list[i].sci; // update the min 33 }

34 if (list[i].min > list[i].lit) { // if lit. is lower 35 list[i].min = list[i].lit; // update the min 36 }

37 }

38 GrandPrize(list); // get grand prize 39 // get math prize

3

40 for (j = 0; j < 100; j++) { // get math score 41 if (list[j].min < 80 && list[j].min >= 60) {

42 score[j] = list[j].math;

43 }

44 else score[i] = 0;

45 }

46 printf("Math Prize:\n"); // print title 47 printScore(score, 1); // print math score 48 // get science prize

49 for (j = 0; j < 100; j++) { // get sci. score 50 if (list[j].min < 80 && list[j].min >= 60) {

51 score[j] = list[j].sci;

52 }

53 else score[i] = 0;

54 }

55 printf("Science Prize:\n"); // print title 56 printScore(score, 1); // print sci score 57 // get literature prize

58 for (j = 0; j < 100; j++) { // get lit. score 59 if (list[j].min < 80 && list[j].min >= 60) {

60 score[j] = list[j].lit;

61 }

62 else score[i] = 0;

63 }

64 printf("Literature Prize:\n"); // print title 65 printScore(score, 1); // print lit. score 66 }

67

68 // to print the grand prize winner

69 // input: struct STU list, a list of students

70 // return: no return

71 // output: print the grand prize winner

72 void GrandPrize(struct STU list[100])

73 {

74 int i; // loop control 75 double score[100]; // array to store score 76

77 // get score as the sum of three subjects

78 for (i = 0; i < 100; i++) {

79 if (list[i].min >= 80) {

80 score[i] = list[i].math + list[i].sci + list[i].lit; 4

81 }

82 else score[i] = 0;

83 }

84 printf("Grand Prize:\n"); // print title 85 printScore(score, 0); // print grand prize 86 }

87

88 // to print the winners of each prize and their score

89 // input: double score, a array of score

90 // int subPrize, whether it's printing a subPrize

91 // return: no return

92 // output: print all score if it's not printing subject prize (subPrize == 0) 93 // print top ten if it's printing subject prize (subPrize == 1) 94 void printScore(double score[100], int subPrize)

95 {

96 int i; // loop control

97 int count = 1; // to count how many winner

98 int i\_max; // store index of maximum score

99 double score\_max; // store the maximum score

100

101 do {

102 score\_max = 0; // initialize the maximum score 103 for (i = 0; i < 100; i++) {

104 if (score\_max < score[i]) { // if current score > maximum score 105 score\_max = score[i]; // update maximum score 106 i\_max = i; // store the index of maximum score 107 }

108 }

109 if (score\_max > 0) { // if maximum score > 0 110 printf("%3d: %s %s %lg\n", // print the winner's 111 count, // sequence 112 list[i\_max].fName,list[i\_max].lName, // name 113 score[i\_max]); // score 114 score[i\_max] = 0; // this winner if found 115 count++; // find next winner 116 }

117 } while (score\_max != 0 && !(subPrize && count > 10));

118 // keep searching until all score if found

119 // if printing the subject prize (subPrize == 1), print at most ten 120 }

5