

CS23710 Worksheet Week Four 2000-01

David Price

February 23, 2001

1 Introduction

This week's practical is intended to provide practice with structures in 'C' together with the use of arrays. The problems in this practical build on those from last week.

PLEASE NOTE: We require you to all work using constructions which are valid in ANSI 'C' and not to use any extensions which the gcc compiler might provide. We suggest that you ALWAYS specify the appropriate flags to gcc to force it to give you warnings etc. if you stray outside of ANSI 'C'.

2 The Problem - Part One

A tri-athlon competition consists of three components, a cycling race, a swim and a final running race. The winner of the competition is the one with the shortest total time for the three races.

This part is identical to part one from last week, but this time you are REQUIRED to use an appropriately specified structure to hold the values being processed by your program. The structure should have a field to hold the competitor's name, a field to hold the cycling race time in seconds, another to hold the swimming race time in seconds, another to hold the running race time in seconds and a final field to hold the total time in seconds.

Write a program to calculate the total time for a single competitor.

Your input should consist of ten items. Firstly, the name of the competitor on a single line; this should not include any "white space", use underscore to provide gaps in the name. A second line of input should provide three integers representing the competitor's time for the cycle race in hours, minutes and seconds. A third line of input should provide the competitor's time for the swim in the same format. A fourth line of input should provide the competitor's time for the final running race in the same format.

The user should be provided with a prompt for each line of input expected.

An example of the input, together with the prompts is:

Competitor's name: Fred_Bloggs
Time for cycle race (hours minutes seconds): 1 35 47
Time for the swim (hours minutes seconds): 1 29 23

Time for the running race (hours minutes seconds): 1 05 14

The program output should be the competitors name and total time in the format shown below.

Competitor Fred_Blogs has a total time of 3Hrs 5Mins 10Secs.

Your input and output should be in EXACTLY the format given above.

Don't forget your previous lectures (in other modules) on algorithm design. Before you start to tackle any computer program, write down a solution on paper, carry out a pencil and paper test, revise the algorithm if necessary. Only when you are happy that you have a working algorithm should you start to write the code.

3 The Problem - Part Two

Your solution to this section should use an array of structures. The structures should be exactly those defined in part one, except that you should add one extra integer field which you use to hold the 'competitor number'.

This problem is very similar to that of part two last week, but this time you are given a fixed number of competitors in advance, rather than reading the data as input from the user.

The competition has precisely eight competitors.

Your program should loop through the eight competitors, requesting the data for each competitor, storing each competitor's details in the array and printing out their total time etc in the format specified in part one.

When the data for all the competitors has been provided, your program should then print out the names and times of all the competitors in a table form.

Don't forget the earlier remark about designing first and don't forget to use meaningful variable names and comments.

4 The Problem — Part Three

You are now required to extend the program from part two so that after printing out the results in the tabular form, your program then sorts all the results in the array until the fastest competitor is stored in element zero. After sorting the data your program should print out the sorted results. The output should look like...

NAME	competitor number	cycle time	swim time	run time	total time
Helen_Fuell	7	1H 15M 10S	1H 47M 17S	1H 05M 27S	4H 07M 54S
Bert_Hill	3	1H 23M 09S	1H 57M 05S	1H 10M 04S	4H 30M 18S

and so on for the other six competitors....

It is very important that you design what you plan to implement for this section before you start to write the code.

NOTE: I want you to write code to implement a simple sorting algorithm, NOT just use a function from the C libraries !

5 Effort Allocated to the Worksheet and Assessment

This week's practical, is the second of a short sequence based around the same problem. For CS23710 students, the final worksheet of this "sports competition" sequence will be assessed as the next contribution to your mark for the module.

You may need to spend an hour or so of your own time to complete this worksheet in addition to your two hour practical.

6 Submission Date (for CS23710 students)

This week's worksheet itself will not be assessed individually. As mentioned above, this worksheet is part of a planned sequence and it is in your own interests to complete the work as it will make the following worksheet less of a challenge.

Next week's worksheet will be the final one of this sequence before we move onto another problem. Next week I will specify that the result of your efforts must be given in. I will not ask for a supporting document, but I WILL expect that the code is internally documented with comments. During assessment of the work, we will carefully consider issues such as layout, naming of variables, general readability, suitability of comments and so on as well as simple accuracy of the code. We expect you to use ANSI 'C' features and we will penalize breaches of this requirement.

The demonstrators will keep a record of who attends the practical.