

CSE 1301 PRACTICAL SESSION 11 Numerical Computing and Search (10 marks)

The aim of this session is to give you some understanding of the impact that numerical considerations have on programming, and also to do some coding involving search algorithms.

Coding style and documentation: You are expected to document your programs and to use a standard, clear, and consistent coding style. Up to 2 of the marks for each practical class may be deducted for poor coding style and/or inadequate documentation.

Preparation (to be completed before class) **(2 marks)**

Write all the algorithms required for the questions below, and attempt the code for all the questions.

PART 1 (Numerical Computing) **(1 mark)**

- (a) Write a program that adds the fraction $1/1000$ one thousand times. This fraction should be stored in a variable of type float. Do you get the result you expect? Why?
- (b) Now store the fraction in a variable of type double. What result do you get? What happens if the fraction is smaller, e.g., $1/1,000,000$?

Supplied information for Parts 2 and 3: Handling student and class data

http://www.csse.monash.edu.au/courseware/cse1301/pracs/prac11_code

This directory contains code that you should use in this prac. This code contains the following.

- (a) A declaration for a struct that stores the information below for each student in a class.
 - a. last name
 - b. first name
 - c. ID number (long int)
 - d. final mark
- (b) A declaration for a struct for a class. This struct contains the structs for the students in the class and the number of students currently enrolled in the class. Note that it is assumed that the maximum class size is `MAXCLASSSIZE` (=10) and the maximum name length is `MAXNAMELEN` (=30).
- (c) A C function that reads from a file information about the students in a class, and stores this information in the class struct. The file has one student record per line.
- (d) A C function that prints the following information about a particular student in tabular form: student ID, first name, last name and final mark. This function receives as a parameter a struct containing the details of the student.
- (e) A C function that prints the number of students in the class, and the following information about each student in tabular form: student ID, first name, last name and final mark. This function receives as a parameter an array of structs containing the student details.

PART 2: Linear Search**(4 marks)**

Write a C function that receives the following parameters: (i) a *search key* (which could be any of the members of the student struct), (ii) a value and (iii) the struct containing the class information, and performs linear search to find the first student whose value matches the search key. The function then calls the `printStudInfo` function described in (c) to print the information about this student. For example, if the search key is "last name" and the value is "Smith", the function returns the details of the first student whose last name is Smith.

The code for the linear search function is available at

http://www.csse.monash.edu.au/courseware/cse1301/lectures/lect31/lect31_examples/

Hints:

- o The search key and the value are character strings; use `sscanf` or `atoi` and `atol` to convert character strings to `int` and `long int`.
- o You may want to use a comparison function which generalizes the `strcmp` function so that it works for all elements of the structure containing character strings, `ints` and `long ints`.

Note: assume that there are four student files, each sorted in ascending order according to a different key. The search key directs the program to read the appropriate file. Sample files may be retrieved from

<ftp.csse.monash.edu.au>

in `subjects/cse1301/Lab11TestDataP3.txt`

PART 3: Binary Search**(3 marks)**

Write a C function that receives the same parameters as the function in PART 2, and performs *binary search* to find a student whose value for the search key matches the given value. When this student is found, his/her details are printed.

The code for the binary search function is available at

http://www.csse.monash.edu.au/courseware/cse1301/lectures/lect31/lect31_examples/

Submission:

Preparation: Show written preparation work to demonstrator at start of class for marking. If not done before class, it will not receive any marks, however you should still show it to your demonstrator during class to check your understanding.

PART 1, 2, 3: To be marked by the demonstrator during this class. No late submission.