Przemyslaw Pawluk, pawluk@cse.yorku.ca

CSE 2031 Software Tools
Assignment 2 – Simple database
Due Date: July 5, 2010 12:00pm

1. Specification

Your programme should store following information about customers using appropriate structure:

- **custId** – customer number, 8 digits
- **Name** – name of the customer, 15 characters
- **Surname** – surname of the customer, 20 characters
- **accBalance** – account balance, signed float

a. Part I – Linked List

Implement linked list providing following operations:

- **`add(head, element)`** — put element to the list
- **`get(head, element)`** —take (remove) the element from the list
- **`empty(head)`** —providing info if the list is empty
- **`print(head)`** – printing the content of the list to the standard output
- **`findName(head, surname)`** — finding element based on surname, and returning pointer to it
- **`findId(head, custId)`** – finding element based on custId, and returning pointer to it.

The records should be stored in a linked list sorted by `custId` (ascending order).

b. Part II – Database

Implement the in-memory database that uses linked list implemented in Part I.

Programme reads the standard input and prints to the standard output. Your programme should provide following functionalities (input corresponding with command is in brackets):

- **Insert** (i or I) – adding new element to the database

User provides the data in the following format:

99999999 name surname (-)[9]9[.99]

Where name and surname are strings without white spaces and (-)[9]9[.99] means float with optional sign. Fields are separated by exactly one space. One line contains one record.

Your program should handle following erroneous input:

- Wrong format, which includes too few or too many digits in custId, too long name or surname, missing field or wrong order. Program should return message Wrong format

If record has been inserted a message Record inserted should be returned.

If record has not been inserted because of a problem with memory allocation message Memory not allocated should be returned.

- **Search** (s or S) – searching for the element based on surname. Program should prompt user to provide surname. If customer exists in the database his record is printed (see print selected). If customer has not been found then message Customer not found should be returned.
- **Remove** (r or R) – deleting the element from the database. Program should prompt user to provide custId. If customer exists in the database user is asked to confirm action (y or Y for yes and n or N for no). If customer does not exist in database then message Customer not found should be returned.
- **Print** (p or P) – print command. Program should ask user if print all or selected
  - **all** (a or A) – printing (to the standard output) content of the database
  - **selected** (e or E) – printing (to the standard output) chosen element. Program should ask user to provide custId
- **Clean** (C) – removing all elements from the database. Program should ask user to confirm (y or Y for yes and n or N for no)
- **Exit** (x or X) – end the execution

Each record should be printed in the following format:

```
|CustId| Surname | Name | Account balance|
```

Name and surname are aligned to max length of the field (15 and 20 respectively). Account balance is aligned to 2 decimal places.

2. Submission

All submitted files should contain following header:

```
/*********************************************
*          CSE2031 - Assignment 2            *
*          File_name                         *
*          Author: Name and Surname          *
*          email: address                    *
*          cs_num:                           *
**********************************************/
```

You program will have four files: **a2.c** file containing main program, **a2.h** file containing all necessary declarations and constants needed in main, **list.c** containing implementation of the linked list, **list.h** providing all necessary declarations and functions to use list in main programme. All files should be placed in directory called a2.

You should submit entire directory **a2**. Your submission should consist of exactly 4 files mentioned above. There is a penalty of 5 points (out of 100) for improper submission like tar files or subdirectories. Your program will be verified against memory leaks (allocate things and not deallocate them later). There is penalty of 10 points for memory leaks.

Use submit command to submit your files

```
submit 2031 a2
```

3. Grading

We will use the following marking scheme

You can be granted up to 100 points

- 30% – linked list
- 60% – calculator
- 10% – coding style
- 5 points penalty for improper submission
- 10 points penalty for memory leaks

4. Sample input and output

```
--------------Begin----------------

I

11111111 John Doe 12.78

P

A

i

222 James Dean 1298.09

P

A

C

y

e

---------------End------------------
```

Output

```
--------------Begin----------------

Record inserted

Print all record (A) or selected record (E)?

|11111111| John            | Doe                 | 12.78|

Wrong format

Print all record (A) or selected record (E)?

|11111111| John            | Doe                 | 12.78|

Clean the database? y/n

---------------End------------------
```