May 2016

98-Comp-A4

Program Design and Data Structures

3 Hours Duration

Notes:

1. If doubt exists as to the interpretation of a question, the candidate is urged to submit with the answer paper a clear statement of any assumptions made.
2. No calculator permitted. This is a Closed book exam.
3. Answer any five of the eight questions.
4. Any five questions constitute a complete paper. Only the first five questions as they appear in your answer book will be marked.
5. For questions that ask the candidate to write a program, pseudocode or any high-level language (e.g. C or C++) is acceptable unless otherwise specified. In all cases, marking will emphasize the operation of the program and not syntactic details.
6. All questions have equal weight.

Marking Scheme

1. (a) 10 marks; (b) 10 marks.
2. (a) 10 marks; (b) 10 marks.
3. 20 marks.
4. (a) 10 marks; (b) 10 marks.
5. (a) 10 marks; (b) 10 marks.
6. 20 marks.
7. 20 marks.
8. 20 marks.

Total mark is out of 100.

Question 1. *Programming.*

(a) Write a program that prompts the user to input a birthdate and responds by printing the horoscope sign corresponding to the birthdate. The birthdate format is month (1-12), followed by a space, then followed by a day (1-31). The program should check for invalid months or invalid days within a month.

Here are some examples:

```
Enter birthdate: 10 18
Sign is Libra

Enter birthdate: 1 12
Sign is: Capricorn

Enter birthdate: 2 30
Invalid birthdate
```

Here are the horoscope signs and their dates:

Aries	March 21 – April 19
Taurus	April 20 – May 20
Gemini	May 21 – June 21
Cancer	June 22 – July 22
Leo	July 23 – August 22
Virgo	August 23 – September 22
Libra	September 23 – October 22
Scorpio	October 23 – November 21
Sagittarius	November 22 – December 21
Capricorn	December 22 – January 19
Aquarius	January 20 – February 18
Pisces	February 19 – March 20

(b) Horoscope signs of the same *Element* are most compatible. There are 4 Elements in astrology and 3 signs in each: FIRE (Aries, Leo and Sagittarius), EARTH (Taurus, Virgo and Capricorn), AIR (Gemini, Libra and Aquarius) and WATER (Cancer, Scorpio and Pisces). One is most compatible with a person with the same sign or the other two signs in the same Element.

Extend your program in part **(a)** to also print the other two signs a birthdate is most compatible with.

Question 2. *Programming.*

(a) An integer is said to be "self-describing" if the following holds. When digit positions are labeled 0 to n-1 from left to right, the digit in each position is equal to the number of times the digit position appears in the number. Thus, the integer 2020 is a 4-digit self-describing integer. Position 0 has value 2 and there are two 0's in the number; position 1 has value 0 and there are no 1's in the number; position 2 has value 2 and there are two 2's; and position 3 has value 0 since there are zero 3's. These are also self-describing integers: 1210 and 3211000.

Write a program that prompts the user for a positive integer and responds by printing a message that indicates if the entered number is self-describing or not.

Hint: you may want to read the digits into an array.

(b) In a crypto-arithmetic puzzle, a mathematical equation is written using letters. Each letter can be a digit from 0 to 9 but no two letters can be the same. This is an example of such a puzzle:

SEND + MORE = MONEY

A solution to this puzzle is D=7, E=5, M=1, N=6, O=0, R=8, S=9, and Y=2.

Write a program to read a crypto-arithmetic puzzle and print a solution to it. For simplicity, assume the puzzle is in the form $x + y = z$, where x and y are each no longer than 8 letters.

Hint: read the input terms of the puzzle into character arrays and use a nested loop for each unique letter in the puzzle.

Question 3. *Object-Oriented Design.*

Sets of numbers are used in many applications. However, some languages, like C++, do not have "Set" as a data type, nor do they directly support set operations.

Design and write a C++ class (call it **Set**) for supporting sets and their operations. Your class should allow for the declaration of sets, both empty and initialized with elements. It should allow for the following set operations: addition of an element, deletion of an element, and checking if an element is a member of the set. Your implementation should allow for sets of various number types (i.e., sets of integers, sets of floats, etc).

You have freedom to select the syntax of the above operations. State any assumption you make clearly. Separate your class into a **Set.h** header file and a **Set.cc** implementation file.

Question 4. *Pointer-based Data Structures.*

An element of a doubly linked list can be defined as follows, expressed in C:

```
typedef struct element {
    int data;
    struct element *prev;
    struct element *next;
} ELEMENT;
```

(a) Write two functions `dlinked_add()` and `dlinked_del()` that add and delete nodes from a *sorted* doubly linked list. The header of each function is shown below. The two functions must work correctly for empty lists.

```
/* Insert a new node onto the list pointed to by
   head, keeping the list sorted  */
void * dlinked_add(ELEMENT *head, ELEMENT *new);

/* Delete the node with the corresponding data value
   from the list pointed to by head, keeping the list
   sorted. If there is more than one node with the
   same data value delete only one. If no node exists
   with the data value, return NULL */
ELEMENT *dlinked_del(ELEMENT *head, int data);
```

(b) Write a function `del_dupl()` that deletes duplicate valued elements in a doubly linked list. The header of the function is shown below. The function must work correctly for empty lists.

```
/* delete duplicate valued elements in the list pointed to
   by head
*/
void del_dupl(ELEMENT *head);
```

Question 5. *Pointer-based Data Structures.*

(a) A node in a binary tree can be defined as follows, expressed in C:

```
typedef struct treenode {
    int data;
    struct element *left;
    struct element *right;
} TreeNode;
```

Write a function `inorder()` that traverses the tree using inorder traversal. The header of the function is shown below. The function must work correctly for an empty tree. Assume at each node, data is printed to the standard output.

```
/* Inorder traversal
*/
void inorder (TreeNode *root);
```

(b) Write a function **`find_max_leaf ()`** that finds the largest data value stored in a leaf of the tree. The header of the function is:

```
/* Find the largest data value in a leaf of the
   tree pointed to by root */
int find_max_leaf (TreeNode *root);
```

Question 6. *File I/O.*

The Securities and Exchange Commission (SEC) has in its possession three files on disk. One lists all employees of firms that rendered financial advice to a particular Wall Street company. The second file has the names of all individuals who traded heavily in that company's stock. The third consists of every name found in the personal Rolodex of a financial advisor, recently convicted of insider trading. Each file is in alphabetical order.

Write a program that helps the SEC search for illegal insider traders by finding and printing names common to all three files.

Assume that names in each file appear one per line, and are all in the same format of: last name, first name. However, the number of lines in each file is not known.

Question 7. *File I/O.*

Write a program to *merge* two sorted input files into one sorted output file. Assume each file has a number of "records", each consisting of a sequence of characters (including white spaces) terminated by a newline character, \n (assume a maximum record size of 80 characters including the terminating newline). The input files may have different numbers of records. Prompt the user for the names of the three files.

It is best that you start your answer with a short paragraph that describes your strategy for solution, the follow with the code. Include comments in your code!

Question 8. *Algorithm Design and Sorting.*

A programmer is given an array A of random integers. The integers in the array are not in any sorted order. However, the array may contain duplicate integers. The programmer must create another array B that contains all the integers in A but without any duplicates. The integers in B need not be in any sorted order. The programmer is contemplating two ways to accomplish this task:

1. For each integer in array A, copy the integer into array B unless the item already exists in B.

2. Sort A using the Quicksort sorting algorithm. Now repeat part (a).

Given that the array contains a large number of integers, which of the above two methods will be faster? Justify your answer.