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 seven of the ten questions.

4. Any seven questions constitute a complete paper. Only the first seven 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. 20 marks.
2. 20 marks.
3. 20 marks.
4. 20 marks.
5. 20 marks.
6. 20 marks.
7. 20 marks.
8. 20 marks.
9. (a) 10 marks; (b) 10 marks.
10. (a) 10 marks; (b) 10 marks.

Total mark is out of 140, which will then be normalized to 100.

Question 1. *Programming.*

An electrical supply company manufactures electrical extension cords rated at 3.0, 6.0, 7.0, 10.0, 13.0, and 15.0 amperes. If an extension cord of length at most 50 feet is to be used with an appliance, a cord with the same amperage rating as the appliance (or higher) can be safely used. However, if the length of the cord is more than 50 but less than 100 feet, a cord of at least the next higher amperage rating is needed. A cord longer than 100 feet should not be used. Write a program to determine and print the proper extension cord selection, given the amperage of the appliance and the cord length. Your output should look something like this:

```
Enter appliance amperage: 5.8
Enter cord length in feet: 60

Use a cord rated at 7 amperes or more.
```

Question 2. *Programming.*

An International Standard Book Number (ISBN) is a code of 10 characters separated by dashes such as 0-670-82162-4. An ISBN consists of four parts: a group code, a publisher code, a code that uniquely identifies the book among the publisher's offerings, and a check character. For the ISBN 0-670-82162-4, the group code is 0, which identifies the book as one from an English-speaking country. The publisher code 670 identifies the book as a Viking Press publication. The code 82162 uniquely identifies the book among the Viking Press publications (Homer: *The Odyssey*, translated by Robert Fagles). The check character is computed as follows:

1. Compute the sum of: the first digit, plus two times the second digit, plus three times the third digit, ... , plus nine times the ninth digit.

2. Compute the remainder of this sum divided by 11. If the remainder is 10, the last character is X; otherwise, the last character is the remainder.

For example, the sum for the above ISBN number is 158. The remainder is 4, which is the last character in the ISBN.

Write a program that prompts the user for an ISBN, reads the candidate ISBN from the standard input into an array, checks whether the candidate ISBN is a valid ISBN, and prints an appropriate message to the standard output.

Question 3. *Programming.*

A problem that often arises in the design of Very Large Scale Integration (VLSI) circuits is to determine whether or not two circuit wires inadvertently intersect. Let's assume that for all practical purposes wires are either vertical or horizontal. A pair of endpoint coordinates $(x_1,y_1:x_2,y_2)$ describes each wire. Hence, (6,8:3,8) is a horizontal wire and (2,4:2,6) is a vertical one. Consequently, a simple check of the x and y coordinates of two wires can reveal if the two wires intersect.

Write a program that reads 20 pairs of coordinates and determines which wires intersect. Your program should print the coordinate pairs of intersecting wires.

Hint: read endpoint pairs into two arrays.

Question 4. *Sorting.*

Insertion sort, Bubble sort, and Quicksort are three popular methods for sorting elements in an array of size *n*. Give the time complexity for each method as a function of *n*. Which of these methods would you use if you had to sort 30 elements? Which would you use if you had to sort 3 million elements? Justify your answer in each case.

Question 5. *Object-Oriented Design.*

Vectors are used in many engineering applications. However, some languages do not have a "vector" data type, nor do they directly support vector operations.

Design and write a C++ class (call it `Vector`) for supporting vectors and their operations. Your class should allow for the declaration of a vector of a given size, without initialization of its elements. Your class should allow for the accessing (read & write) of elements of a vector. It should also allow for the addition, subtraction, multiplication, and printing of vectors. These operations should produce an error message if the vectors involved in an operation are not of the same size.

Use C++ templates to support vectors of various types (only assume `int`, `float`, and `double` for simplicity). Overload the usual arithmetic operators to provide addition, subtraction, and multiplication of two vectors. Also overload the equality operator to allow equality comparison of two vectors.

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

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.*

A **palindrome** is a sequence of characters that reads the same forwards as backwards. For example, "bob", "deed" and "level" are palindromes, while "test" and "Hello" are not.

Write a **recursive** function: `int palindrome (char *seq)` that returns 1 if the sequence of n characters pointed to be `seq` is a palindrome, and returns 0 otherwise.

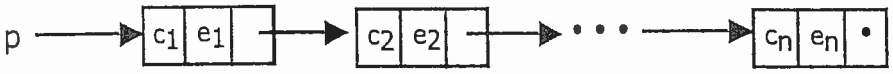
Your solution should not exceed a few lines of code.

Question 9. *Linked Lists.*

Consider the following definition of a `polynomial_node` structure expressed in `C`.

```
typedef struct {
    int coefficient;
    int exponent;
    polynomial_node *next;
} polynomial_node;
```

A polynomial $p(x) = c_1x^{e_1} + c_2x^{e_2} + \cdots + c_nx^{e_n}$ may be represented as a linked list of `polynomial_nodes`, as shown below:



where $c_1, c_2, \ldots, c_n > 0$ and $e_1 > e_2 > \ldots > e_n \geq 0$ are all integers.

(a) Write a function `polynomial_node * get_polynomial()` that reads pairs of coefficient-exponent entries `(c,e)`, creates a linked list representing the corresponding polynomial, and returns a pointer to the head of the newly created list. Assume the input pairs sorted by exponent values, and are terminated by a `(0,0)` entry.

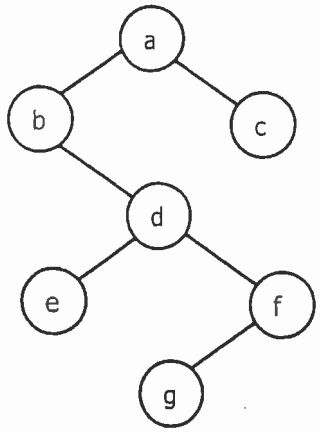
(b) Write a function:

```
polynomial_node * add_polynomials (polynomial_ node * p1,
                                   polynomial_ node * p2)
```

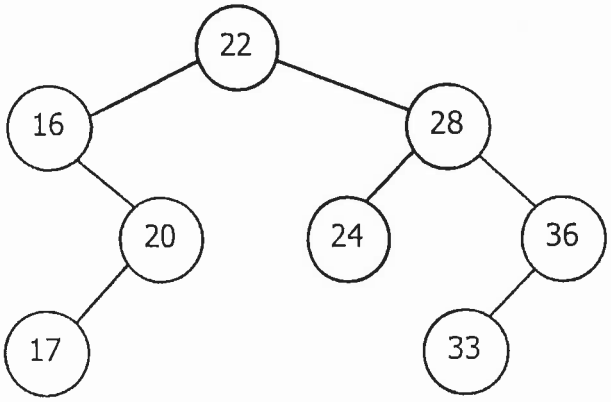
that adds two the polynomials represented by the lists pointed to by `p1` and `p2` respectively. The resulting polynomial is represented by a newly created list, a pointer to which is returned by the function.

Question 10. *Binary Trees.*

(a) Give the *inorder*, *preorder* and *postorder* traversals of the tree shown below.



(b) In a Binary Search Tree (BST), the key of each node is always smaller than the key of its right child and always greater than the key of its left child. The tree below is a BST.



Show the binary search tree after a node with key "34" is inserted.

Show the binary search tree after the node with key "33" is deleted.

Show the binary search tree after a new node with key "33" is inserted.