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 six of the nine questions.

4. Any six questions constitute a complete paper. Only the first six 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.

Question 1. *Programming.*

(a) Genius, according to Thomas Edison, is a matter of 2% inspiration and 98% perspiration. Nowadays, perspiration is usually socially unacceptable, except for men and horses. Alternatives—such as sweating, glowing and dripping—exist, and it is important to know when to use them, regardless of the consequences to genius. According to my social secretary, the rules are:

Temperature	Cooling Mechanism
Over 100°	Dripping
91° to 100°	Sweating
81° to 90°	Perspiring
71° to 80°	Glowing
70° or below	Gleaming

Write a program to read a temperature and let you know what cooling mechanism to use.

(b) Three desperados stole a shipment of gold bars late one night. After escaping to their hideout, they resolved to divide the booty in the morning and went to bed. However, as soon as one of the bandits heard the others snoring, he split the stolen gold into three equal piles, and found one bar left over. He buried one of the three piles under a tree, along with the extra bar, and left the rest of the gold. Then he went to sleep, sure that he had protected his interest in the treasure. Naturally, the other two outlaws were no more honest than the first. Each in turn crept to the cache of gold, divided it three ways, and found one bar left over, which he kept along with "his third".

Soon came the morning and the final three-way division. Oddly enough, this division also left one bar remaining. The highwaymen fought over this bar, and in an unprecedented three-way draw, shot each other dead.

Write a program to find how many bars could have been in the original loot. Assume that the loot contained no more than 500 bars.

Hint: Write a subprogram to determine if the above divisions of the gold are possible for a given number of bars and then use it.

Question 2. *Programming.*

Consider the scene in a closed-room secret design session for the 2005 model of a big-name automobile. All around the conference table are engineers and executives trying to strike a balance between greed (which leads them to build the cheapest car possible) and fear (which pulls them in the other direction, towards more reliable cars).

The relation between the mean time between failure (MTBF) of a system and its subsystems can be described as:

$$\frac{1}{MTBF_{total}} = \frac{1}{MTBF_{sub1}} + \frac{1}{MTBF_{sub\,2}} + \cdots + \frac{1}{MTBF_{subn}}$$

and is correct if the failure of any subsystem dooms the entire car.

Assume that the costs and MTBF's of various subsystems are:

Subsystem		MTBF	Manufacturing Cost
brakes	disc	4 years	$15.00
	drum	5 years	$25.00
engine	wankle	3 years	$1,067.00
	conventional	6 years	$1,850.00
suspension	air	9 years	$430.00
	oil	7 years	320.00
electrical	computer	2 years	130.00
	standard	4 years	$40.00

Write a program that models each combination of subsystems and answers the following questions:

1. Which system is the cheapest?
2. Which has the longest MTBF?
3. Which system has the lowest cost per failure-free year?

Hint: use nested loops.

Question 3. *Programming.*

A **magic square** is a square array of integers such that the sum of every row, the sum of every column and the sum of each of the two diagonals are all equal. This is an example of a 4x4 magic square.

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

Write a program that reads the size of the square n and allocate an nxn two-dimensional array of integers. The program then reads n^2 integers that represent the rows of the square from top to bottom. The program finally determines whether or not the square is a magic square.

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.*

Complex numbers occur in many engineering applications. However, most languages, including **C++** and **Java**, do not have "complex number" as a data type, nor do they directly support complex arithmetic.

Design and write a **C++** class (call it **Complex**) for supporting complex numbers and their arithmetic. Your class should allow for the declaration of complex numbers with and without initialization of real and imaginary parts. It should allow for the accessing (read & write) of the real and imaginary parts of a complex number. It should also allow for the addition, subtraction, multiplication, and printing of complex numbers.

Separate your class into a **Complex.h** header file and a **Complex.cc** implementation file.

Question 6. *File I/O.*

Run-length encoding is a data compression technique that exploits repetition to shorten overall length. A marker and a count replace any series of identical characters. For example, the following sequence of characters in a file

```
aaaaabbccccdddefg
```

can be replaced by `\5abb\4c\3defg`. Note that the replacement of `a` and `c` did save space, while that of `d` had no effect and that `b`, `e`, `f` and `g` were not replaced since compressing them would actually waste space.

Write a program that reads characters from a file one at a time, and uses run-length encoding to write a compressed file. Make your program as efficient in time and space as possible.

Question 7. *File I/O.*

Write a program to generate personalized mail. The program takes input both from an input file and from the keyboard. The input file contains the text of a letter, except tat the first name of the recipient is indicated by the three characters `#N#`. The program prompts the user for a name and then writes the letter to a second file, but with the three characters replaced by the name. The three characters `#N#` may occur more than once in the file.

Question 8. *Linked Lists and Trees.*

(a) 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;
```

Write a function `del_dupl()` that deletes duplicate valued elements in a doubly linked list. The prototype 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);
```

(b) 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 prototype 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);
```

Question 9. *Program Design.*

In an ancient land, the beautiful princess Eve had many suitors. She decided on the following procedure to determine which suitor she would marry. First, all of the suitors would be lined up one after the other and assigned numbers. The first suitor would be number 1, the second number 2, and so on up to the last suitor, number *n*. Starting at the suitor in the first position, she would then count three suitors down the line (because of the three letters in her name) and the third suitor would be eliminated from winning her hand and removed from the line. Eve would then continue, counting three more suitors, and eliminate every third suitor. When she reached the end of the line, she would continue counting from the beginning.

For example, if there were 6 suitors, the elimination process would proceed as follows:

123456	initial list of suitors, start counting from 1
12456	suitor 3 eliminated, continue counting from 4
1245	suitor 6 eliminated, continue counting from 1
125	suitor 4 eliminated, continue counting from 5
15	suitor 2 eliminated, continue counting from 5
1	suitor 5 eliminated, 1 is the lucky winner

Write a program that creates a circular linked list of nodes to determine which position you should stand in to marry the princess if there are *n* suitors. Your program should simulate the elimination process by deleting the node that corresponds to the suitor that is eliminated for each step in the process.