

University of Swaziland
Department Of Computer Science
Final Examination 2006-2007

Title of paper: *C under Unix*
Course number: *CS344*
Time Allowed: *Three (3) hours*

Instructions:

- *Each question is worth 25 marks*
- *Answer all questions in section A.*
- *Answer any two (2) questions in section B.*

This paper may not be opened until permission has been granted by the invigilator

Section A

- Answer both questions 1 & 2 below.

Question 1-25 marks

(Compulsory)

Using UML notation, draw a class diagram for the VIP cosmetics company described below. Your diagram must show suitable classes, association and processing dependencies, and accessibility of class members.

VIP cosmetics

VIP cosmetics is a successful network sales company. The company sells a variety of health and beauty products through selected distributors. A distributor is a person who has committed himself to selling the products of the company. The distributors then sell to the general public. VIP keeps a list of all its distributors. This list details the distributor name, postal address and other essential details.

Distributors buy products by placing orders with VIP head office. The order contains a list of items that the distributor wants to buy. This list details the product identity code and the quantity being ordered. The total cost for each order item may also be computed by multiplying the quantity by the product price. The product price may be obtained from a list of all the products. This list details the product code, name, description and price. The total cost of each order may be computed by summing the cost of each item on the order.

Distributors receive a discount depending on how many points they have accumulated since joining VIP as a distributor. The number of points is proportional to the accumulated total cost of their orders. For example, if they place an order costing E200 they receive an extra 200 points. Distributors who have just joined VIP receive 30% discount. Distributors who have accumulated 2 500 points receive 35% discount, 10 000 points receive 40% discount, 20 000 points receive 44% discount, 40 000 points receive 48% discount and 80 000 points receives 52% discount.

Question 2 - 25 marks

(Compulsory)

Translate the following Pascal program into an equivalent C++ program.

```
program Stats (Infile);
var InFile : Text;
    Char_count, Vowel_count, Line_count : Integer;
    ch : char;
begin
    Char_count := 0;
    Vowel_Count := 0;
    Line_Count := 0;

    Assign (Infile, 'letter.txt');
    Reset (InFile);

    while not eof (InFile) do
    begin
        while not eoln (InFile) do
        begin
            read (InFile, ch);
            char_count := char_count + 1;
            if ch in ['a','e','i','o','u','A','E','I','O','U'] then
                Vowel_count := Vowel_count + 1;
            end;
            Line_count := Line_count + 1;
            readln (Infile);
        end;
        close (InFile);
        writeln ('Number of characters = ', Char_count:5);
        writeln ('Number of Vowels      = ', Vowel_count:5);
        writeln ('Number of lines          = ', Line_count:5);

        readln;
    end.
end.
```

Section B

- Answer any two (2) questions from questions 3 to 5 below.

Question 3 - 25 marks

Re-write the following C++ program as follows:

- use a **class template definition** instead of **struct** definition of the **Queue** data structure.
- Rewrite the function **void Create (Queue&)** as a constructor for the class.
- Rewrite the function **void ShowAll (Queue)** function as a **friend insertion operator (<<)**
- The whole program must be rewritten to reflect the changes proposed above.

```
#include <iostream>

const int MaxQueue = 10;
typedef int Datatype;
struct Queue
{
    Datatype Items [MaxQueue];
    int Front, Rear;
    int Count;
};

void Create (Queue&);
bool IsEmpty (Queue);
bool IsFull (Queue);
void Enqueue (Datatype, Queue&);
void Dequeue (Queue&);
Datatype QueueFront(Queue);
void ShowAll (Queue);

void main()
{
    Queue Q;
    Create (Q);
    Enqueue (30, Q); Enqueue (10, Q); Enqueue (5, Q); Enqueue (20, Q);
    Enqueue (70, Q);
    ShowAll (Q);
    Dequeue (Q);
    ShowAll (Q);
    Dequeue (Q);
    ShowAll (Q);

    cout << endl;
}
}
```

```

void Create (Queue& Q)
{
    Q.Front = 0;
    Q.Rear = MaxQueue-1;
    Q.Count = 0;
}

bool IsEmpty (Queue Q)
{
    return (Q.Count == 0);
}
bool IsFull (Queue Q)
{
    return (Q.Count == MaxQueue);
}

void Enqueue (Datatype NewElement, Queue& Q)
{
    if (!(IsFull(Q)))
    {
        Q.Rear = ((Q.Rear+1) % MaxQueue);
        Q.Items[Q.Rear] = NewElement;
        Q.Count = Q.Count + 1;
    }
}

void Dequeue (Queue& Q)
{
    if (!(IsEmpty(Q)))
    {
        Q.Front = (Q.Front % MaxQueue) + 1;
        Q.Count = Q.Count - 1;
    }
}

Datatype QueueFront (Queue Q)
{
    return (Q.Items[Q.Front]);
}

void ShowAll (Queue Q)
{
    Queue Temp = Q;
    while (!(IsEmpty(Temp)))
    {
        cout << QueueFront(Temp) << "\t";
        Dequeue(Temp);
    }
    cout << endl;
}

```

Question 4- 25 marks

- a) Distinguish between a function prototype and a function signature. *2 Marks*
- b) Explain the difference between overloading and overriding. *2 Marks*
- c) Explain the difference (if any) between a function template and a template function. *1 Mark*
- d) An integer is said to be perfect if the sum of its factors, including 1(but not the number itself), is equal to the number. For example, 6 is a perfect number because $6 = 1 + 2 + 3$. Write a function **perfect** that determines whether a parameter **number** is a perfect number. Use this function in a program that determines and prints all perfect numbers between 1 and 1000. *20 marks*

Question 5- 25 marks

- a) What is a recursive function? *1 Mark*
- b) Explain the two components (cases) of a recursive definition and how they are related? *3 Marks*
- c) Using C++ notation, write a *non-recursive* linear search function
bool LinearSearch (int A[], int keyValue, int ArraySize)
that takes an array containing *ArraySize* elements and returns **true**, if *keyValue* is an element of the array, and **false** otherwise. *7 Marks*
- d) Write a *recursive* definition of the **LinearSearch** function explained in part c) above. *7 Marks*
- e) Write a *recursive* **PrintArray** function
Void PrintArray (int A[], int StartIndex, int EndIndex)
that displays the elements of an array from startIndex to EndIndex. *7 Marks*

End of Examination Paper