

Paper 3F6: Software Engineering and Design

UI Design and Software Management

Examples Paper 4

Straightforward questions are marked †

*Tripos standard (but not necessarily Tripos length) questions are marked **

User Interface Design

1. You are a member of a team designing a web interface for an internet banking service where each client has two accounts: a current account and a savings account. Assuming that the user has logged-in and passed all security checks, write a specification for each of the following use cases:
 - (a) transfer £100 from the current account to the savings account
 - (b) pay a bill of £36.50 to BT plc, customer number EA3482828
 - (c) close all accounts and transfer any outstanding balance to an account at another bank.
2. A prototype of the internet banking system described in Q1 has been implemented:
 - (a) describe how you would conduct a usability test?
 - (b) what metrics would you use and how would you measure them?

Software Management

3. * As a member of a code review team you are supplied with the information shown in Fig. 1.
 - (a) What is the difference between a *code inspection* and a *walk-through*?
 - (b) List four distinct cases that should be considered when reviewing the operation of the `insert` function.
 - (c) Perform a walk-through of the `insert` function and write a brief report describing your findings.
 - (d) If the integer numbers i were known to lie in the range $0 < i < N$, suggest how the `insert` function could be simplified?

“The supplied C code fragment is intended to implement a linked list of integers stored in ascending order. Each element of the list is a `struct` of type `Item` holding the integer value, a pointer `pred` to the previous element, if any, and a pointer `succ` to the succeeding element, if any. The variable `head` points to the first element in the list, and `tail` points to the last element. Initially, both `head` and `tail` are `NULL`. The function `insert` is intended to insert its argument `x` into the list. If `x` is already in the list, `insert` should do nothing.”

```

struct Item {
    int value;           // the integer value
    Item *succ;         // succeeding value
    Item *pred;         // preceding value
};
Item *head, *tail;     // Head and Tail of List

// Create a new item and return a pointer to it
Item*NewItem(int value, Item *succ, Item *pred);

// Insert x into linked list
void insert(int x)
{
    Item *p = head , *q;
    if(!p) {
        head = tail = NewItem(x,NULL,NULL);
    }else if (x>tail->value){
        tail->succ = NewItem(x,tail,NULL);
        tail = tail->succ;
    }else{
        while (x > p->value) p++;
        if (x==p->value) return;
        q = NewItem(x,p,p->pred);
        p->pred = q;
    }
}
}

```

Figure 1: The insert function

4. The C++ function below claims to efficiently locate a specific integer value in a large array of integers, provided that the array elements are stored in ascending order.

```
// Binary search routine:
// the elements of a[i] i=1..N are stored in sort order.
// return index i of array element a[i] == x
// return 0 if not found
int binary_search(int *a, int x, int N)
{
    int mid, low = 1, high = N;
    a[0] = x;
    do {
        mid = (low+high) / 2;
        if (low>high)
            mid = 0;
        else if (a[mid]<x)
            low = mid + 1;
        else
            high = mid - 1;
    } while (a[mid] != x);
    return mid;
}
```

Assuming that $N=100000$, list the specific cases that should be covered when testing this routine and design a test harness suitable for use as a regression test.

Steve Young
February 2008