

ISOM Project Cluster 2005

Exercise Sheet 1

Søren Debois
debois@itu.dk

September 5, 2005

Here are basic exercises in pointer manipulations. Feel free to modify the exercises as you see fit. Also, please e-mail me if you have any questions, or feel that the exercises below are not enough.

Exercise 1. Give your own implementation of `strlen`, `strcmp`, `strncmp`, `strcat`, `strcpy`, `strcoll`, `strncat`, `strncpy`, `strncoll` (or as many as you find interesting). If you don't know what these functions do, either login to `ssh.itu.dk` and use the `man` tool (e.g., `man strncpy`) or use google <http://www.google.dk/search?q=strlen+manual+page&sourceid=mozilla-search&start=0&start=0&ie=utf-8&oe=utf-8&client=firefox-a&rls=org.mozilla:en-US:official>.

Exercise 2. This class implements binary trees:

```
class Node {
public:
    Node* left;
    Node* right;
    int k;
};
```

By convention, leaves have both `left` and `right` set to null. Implement standard binary search on such trees, assuming all integers in the `left` branch are less than or equal to k , and all integers in the `right` branch are greater than k . Implement a function that inserts a new number maintaining that invariant. Implement sorting of zero-terminated integer-arrays using your above two solutions.

Exercise 3. Implement a circular buffer of integers by adding methods to this class:

```
class CBuf {
    char* q;
};
```

Exercise 4 (Advanced). Implement RC5-encryption/decryption. (Refer to <http://theory.lcs.mit.edu/~rivest/Rivest-rc5rev.pdf> for details on RC5.)

Exercise 5. Modify the `Stack` example of this week and last such that it takes the size of the stack as a constructor argument. Further modify it (perhaps using part of the solution to exercise 1) to grow the stack when needed.