McMaster University
Department of Computing and Software
Dr. W. Kahl

SFWR ENG 2S03
Exercise Sheet 5

SFWR ENG 2S03 — Principles of Programming

11 October 2006

Exercise 5.1 — Top 10 Lists (55% of Midterm 3, 2003)

A computer game maintains its **top 10 list** in **two** arrays, declared **globally** by:

```
#define TOPLEN 10
int top10scores[TOPLEN];
char * top10names[TOPLEN];
```

(These are global arrays and need not be passed as arguments to the functions below.)

Scores in this game are always non-negative, so negative entries in *top10scores* indicate *empty* positions, i.e., positions that have not been claimed yet. For example, after the first player plays this game, achieving the non-negative score s_1 only the entry for the "top position" will be occupied by s_1; all other entries in the array *top10scores* will be negative.

Players who provide their name will have their name listed in the array *top10names* in the same positions that their scores occupy in *top10scores*. Players can play ***anonymously***; instead of their names, the *NULL* pointer value will be stored in their positions in the array *top10names*.

(a) $\boxed{\approx 10\%}$ Some possible states of the two arrays *top10scores* and *top10names* make no sense. For example, there should be no "empty" entries between real scores.

Define precisely which states of the two arrays *top10scores* and *top10names* you consider as legal, and how you interpret these states. In particular, where will the best score be stored?

(b) $\boxed{\approx 20\%}$ **Define the interface** of a function *insertIntoTop* that attempts to insert a new score into the top 10 lists — it will insert only if the new score deserves it, and it will inform the caller of the following:

- whether insertion was successful,
- whether the score of a **different** non-anonymous player was expunged from the list, and if yes, who this was, and what their score was (so the system can, for example, send them an e-mail to ask them to play again),
- the difference between the supplied score and the previous best score.

Document how the caller of the function *insertIntoTop* will be able to access all this information after a call, and **document** the arguments the function *insertIntoTop* accepts and which assumptions it makes about those arguments. — **Hint:** Pass-by-reference may be useful.

(c) $\boxed{\approx 25\%}$ Implement the function *insertIntoTop* from (b).

(d) (***not on the original midterm — independent*** from (b) and (c))

Implement the function *displayTop10* that produces a sensible display of legal states — see (a) — of the top 10 list.

(e) (***not on the original midterm***) Implement an appropriate *main* program to test your functions.

Exercise 5.2 — Find Errors (15% of Midterm 3, 2003)

Find and describe the error in each of the following program segments. If the error can be corrected, explain how.

(a) char $*s$;
printf("%s\n", s);

(b) char s[] = "Some string.";
printf("%s\n". &s[1]);

(c) float $*\, x, y$;
$x = y$;

(d) char s[4] = {’a’, ’b’, ’d’, ’e’};
printf("%s\n", s);

(e) int $z = 5$;
int $*\, p, q$;   /* integer pointers p and q */
$p = \&z$;
$q = *p$;

Exercise 5.3 — Typing (8% of Midterm 2, 2004)

Let the following declarations bee given:

char z[100];
char $*\, c$[15];
int $**\, p$;

Give the types of the following expressions:

(a) $p[42]$

(b) $z + 4$

(c) $*(c+5)$

(d) $\&(c[1])$