

# Princeton University

## COS 217: Introduction to Programming Systems

### Spring 2007 Midterm Exam Answers

The exam was a 50-minute, open-book, open-notes exam. It had one question.

#### Question 1

```
#include "symtable.h"
#include <stdio.h>
#include <assert.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>

struct Node {
    char *phrase;
    struct Node *next;
};

char *extract_acronym(char *phrase) {
    int i = 0;
    int j = 0;
    int capcount = 0;
    char *acronym;

    assert(phrase != NULL);

    /* count how many capital letters are in the phrase */
    while (phrase[i] != '\0') {
        if (isupper(phrase[i]))
            capcount++;
        i++;
    }

    /* make the acronym out of just the capital letters */
    acronym = (char*)malloc(capcount + 2);
    assert(acronym != NULL);
    i = 0;
    while (phrase[i] != '\0') {
        if (isupper(phrase[i])) {
            acronym[j] = phrase[i];
            j++;
        }
        i++;
    }
    acronym[j] = '\n';
    acronym[j+1] = '\0';
    return acronym;
}

void add_a_phrase(SymTable_T table, char *phrase) {
    struct Node *newnode;
    struct Node *phrases;
    char *acronym;

    assert(table != NULL);
    assert(phrase != NULL);

    acronym = extract_acronym(phrase);
```

```

/* Optional: Disallow empty acronyms. */
if (strcmp(acronym, "\n") == 0)
    return;

newnode = (struct Node*)malloc(sizeof(struct Node));
assert(newnode != NULL);
newnode->phrase = (char*)malloc(strlen(phrase) + 1);
assert(newnode->phrase != NULL);
strcpy(newnode->phrase, phrase);

phrases = SymTable_get(table, acronym);
if (phrases == NULL)
    newnode->next = NULL;
else
    newnode->next = phrases;
SymTable_put(table, acronym, newnode);
}

void do_a_query(SymTable_T table, char *acronym) {
    struct Node *phrases;

    assert(table != NULL);
    assert(acronym != NULL);

    phrases = SymTable_get(table, acronym);
    while (phrases != NULL) {
        printf("%s", phrases->phrase);
        phrases = phrases->next;
    }
}

enum {MAXLINE = 1000};

int main(int argc, char **argv) {
    SymTable_T table = SymTable_new();
    char line[MAXLINE];
    FILE *phrases = fopen("phrases", "r");
    assert(phrases != NULL);
    while (fgets(line, MAXLINE, phrases))
        add_a_phrase(table, line);
    while (fgets(line, MAXLINE, stdin))
        do_a_query(table, line);
    /* we will not worry about freeing things */
    return 0;
}

```

Copyright © 2007 by Robert M. Dondero, Jr.