

# COP 3514 #3

David Maldonado

July 10, 2014

---

```
1. #include <stdio.h>                                hw3_string_case.c
   #include <string.h>

   // warning: modifies passed string!
   void upcase(char* text) {
       int length = strlen(text);

       for(int i = 0; i < length; i++) {
           if((text[i] > 96) && (text[i] < 123)) {
               text[i] = text[i] - 'a' + 'A';
           }
       }
   }

   // warning: modifies passed string!
   void downcase(char* text) {
       int length = strlen(text);

       for(int i = 0; i < length; i++) {
           if((text[i] > 64) && (text[i] < 91)) {
               text[i] = text[i] + 'a' - 'A';
           }
       }
   }

   // takes string input and prints same string upcased and downcased
   int main() {
       char s[100];

       printf("Enter string: ");
       scanf("%[^\n]*c", s);
       upcase(s);
       printf("String upcased: %s \n", s);
       downcase(s);
       printf("String downcased: %s \n", s);
```

```
return 0;
}
```

---

```
[dmaldonado1@c4lab02]~/COP3514% git pull
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 6 (delta 1), reused 0 (delta 0)
Unpacking objects: 100% (6/6), done.
From https://github.com/dave-maldonado/COP3514
   9b44a4b..36ce5fa  master    -> origin/master
Updating 9b44a4b..36ce5fa
Fast-forward
 hw2_solutions.pdf | Bin 0 -> 567763 bytes
 hw3_string_case.c | 37 ++++++
 2 files changed, 37 insertions(+), 0 deletions(-)
 create mode 100644 hw2_solutions.pdf
 create mode 100644 hw3_string_case.c
[dmaldonado1@c4lab02]~/COP3514% gcc -std=c99 -Wall -o hw3_string_case hw3_string_case.c
[dmaldonado1@c4lab02]~/COP3514% ./hw3_string_case
Enter string: The independence day of U.S.A is July 4th!
String upcased: THE INDEPENDENCE DAY OF U.S.A IS JULY 4TH!
String downcased: the independence day of u.s.a is july 4th!
[dmaldonado1@c4lab02]~/COP3514%
```

Figure 1: solution for problem 1 compiling and running

---

```
2. #include <stdio.h>
#include <string.h>
#define LINE_LEN 1024

// returns integer given string representation
int strtoint(char str[]) {
    int offset = 0;
    int sign = 1;
    int num = 0;

    if(str[0] == '-') { offset = 1; sign = -1; }
    if(str[0] == '+') { offset = 1; }
    for(int i = offset; i < strlen(str); i++) {
        num = num * 10 + (str[i] - '0');
    }
    return num * sign;
}

// test program that does N string to int conversions and sums them
// NOT safe from overflow!
int main() {
    const int N_CONVERSIONS = 4;
    char buffer[LINE_LEN];
    int sum = 0;

    for(int i = 0; i < N_CONVERSIONS; i++) {
        fgets(buffer, LINE_LEN, stdin);
        buffer[strlen(buffer) - 1] = '\0';
        sum += strtoint(buffer);
    }
    printf("The sum is: %d \n", sum);
    return 0;
}
```

---

```
129 [dmaldonado1@c4lab02]~/COP3514% git pull
130 remote: Counting objects: 5, done.
131 remote: Compressing objects: 100% (1/1), done.
132 remote: Total 3 (delta 2), reused 3 (delta 2)
133 Unpacking objects: 100% (3/3), done.
134 From https://github.com/dave-maldonado/COP3514
135     83f258f..cc90a74  master    -> origin/master
136 Updating 83f258f..cc90a74
137 Fast-forward
138   hw3_convert_string.c |      6 +++--
139   1 files changed, 3 insertions(+), 3 deletions(-)
140 [dmaldonado1@c4lab02]~/COP3514% gcc -std=c99 -Wall -o hw3_convert_string
141 hw3_convert_string.c
142 [dmaldonado1@c4lab02]~/COP3514% ./hw3_convert_string
143 1
144 12
145 123
146 1234
147 The sum is: 1370
148 [dmaldonado1@c4lab02]~/COP3514% □
```

Figure 2: solution for problem 2 compiling and running

---

```

3. #include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
#define LINE_LEN 1024

bool startsWithb(char str[]) { return str[0] == 'b'; }

// test program to take N string inputs and print those starting with 'b'
int main() {
    const int N_STRINGS = 10;
    char buffer[LINE_LEN];
    char* strings[N_STRINGS];

    for(int i = 0; i < N_STRINGS; i++) {
        fgets(buffer, LINE_LEN, stdin);
        buffer[strlen(buffer) - 1] = '\0';
        strings[i] = malloc(strlen(buffer) + 1);
        strcpy(strings[i], buffer);
    }
    for(int i = 0; i < N_STRINGS; i++) {
        if(startsWithb(strings[i])) { printf("PRINT: %s \n", strings[i]); }
    }
    return 0;
}

```

---

```

220 [dmaldonado1@c4lab02]~/COP3514% gcc -std=c99 -Wall -o hw3_strings_with_b hw3_strings_with_b.c
221 [dmaldonado1@c4lab02]~/COP3514% ./hw3_strings_with_b
222 foo
223 bar
224 beagle
225 baby
226 wow
227 doge
228 axolotl
229 fish
230 minecraft
231 pilgrim
232 PRINT: bar
233 PRINT: beagle
234 PRINT: baby
235 [dmaldonado1@c4lab02]~/COP3514%

```

Figure 3: solution for problem 3 compiling and running

---

```

4. #include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
#define LINE_LEN 1024

bool endsWithEd(char str[]) {
    return ((str[strlen(str) - 2] == 'e') && (str[strlen(str) - 1] == 'd'));
}

// test program to take N string inputs and print those ending with 'ed'
int main() {
    const int N_STRINGS = 10;
    char buffer[LINE_LEN];
    char* strings[N_STRINGS];

    for(int i = 0; i < N_STRINGS; i++) {
        fgets(buffer, LINE_LEN, stdin);
        buffer[strlen(buffer) - 1] = '\0';
        strings[i] = malloc(strlen(buffer) + 1);
        strcpy(strings[i], buffer);
    }
    for(int i = 0; i < N_STRINGS; i++) {
        if(endsWithEd(strings[i])) { printf("PRINT: %s \n", strings[i]); }
    }
    return 0;
}

```

---

```

219 Unpacking objects: 100% (3/3), done.
220 [dmaldonado1@c4lab02]~/COP3514% gcc -std=c99 -Wall -o hw3_strings_with_ed hw3_strings_with_ed.c
221 [dmaldonado1@c4lab02]~/COP3514% ./hw3_strings_with_ed
222 laughed
223 laugh
224 long
225 belonged
226 rabbit
227 orange
228 yelled
229 hi
230 young
231 tounge
232 PRINT: laughed
233 PRINT: belonged
234 PRINT: yelled
235 [dmaldonado1@c4lab02]~/COP3514% █

```

Figure 4: solution for problem 4 compiling and running

5. Something interesting happened with this problem. My solution compiles correctly in both clang on my macbook and gcc on the c4lab machine but on the c4lab machine I get stray character codes in the output. A bug in the compiler perhaps? I've included two screenshots after the code.

```
hw3_convert_date.c

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define LINE_LEN 1024

char* convertMonth(int month, char month_ret[]) {
    char* allMonths[] = {"January", "February", "March", "April",
                        "May", "June", "July", "August", "September",
                        "October", "November", "December"};
    return month_ret = allMonths[month - 1];
}

// convert date from 'mm/dd/yyyy' format to 'month-name dd, yyyy'
char* convertDate(char date[], char date_ret[]) {
    char input_month[3];
    char input_day[3];
    char input_year[5];
    char converted_month[9];
    char* converted_month_p;

    memcpy(input_month, &date[0], 2);
    memcpy(input_day, &date[3], 2);
    memcpy(input_year, &date[6], 4);
    input_month[2] = '\0';
    int month = atoi(input_month);
    converted_month_p = convertMonth(month, converted_month);
    strcpy(date_ret, converted_month_p);
    strcat(date_ret, " ");
    strcat(date_ret, input_day);
    strcat(date_ret, ",");
    strcat(date_ret, " ");
    strcat(date_ret, input_year);
    return date_ret;
}

// test program for date format conversion
// NO ERROR CHECKING! enter date as mm/dd/yyyy
int main() {
    char buffer[LINE_LEN];
    while(fgets(buffer, LINE_LEN, stdin) != NULL) {
```

```

    char tempDate[18];
    buffer[strlen(buffer) - 1] = '\0';
    printf("%s \n", convertDate(buffer, tempDate));
}
return 0;
}

```

---

```

360  λ Davids-MacBook-Pro COP3514 → λ git master* → make
361  c99 -g -Wall -O3      hw3_convert_date.c  -o hw3_convert_date
362  λ Davids-MacBook-Pro COP3514 → λ git master* → ./hw3_convert_date
363  01/10/2014
364  January 10, 2014
365  10/13/1999
366  October 13, 1999
367  12/20/1930
368  December 20, 1930

```

Figure 5: solution for problem 5 compiling and running with clang on macbook

```

359  hw1_var_swap.c      hw2_triangle_sides.c  hw3_strings_with_ed.c
360  [dmaldonado1@c4lab02]~/COP3514% gcc -Wall -o hw3_convert_date hw3_convert
361  _date.c
362  [dmaldonado1@c4lab02]~/COP3514% ./hw3_convert_date
363  01/10/2014
364  January 10\323 \377^?, 2014\377^?
365  10/10/2014
366  October 10\323 \377^?, 2014\377^?
367  12/12/1999
368  December 12\323 \377^?, 1999\377^?

```

Figure 6: solution for problem 5 compiling and running with gcc on c4lab machine



---

6. (a) 

```
typedef struct inventory {  
    char partName[30];  
    int partNumber;  
    double price;  
    int stock;  
    int reorder;  
}inventory;
```

---

(c) 

```
typedef struct address {  
    char streetAddress[25];  
    char city[20];  
    char state[3];  
    char zipCode[6];  
}address;
```

---

(d) 

```
typedef struct student {  
    char firstName[15];  
    char lastName[15];  
    address address;  
}student;
```

---

(e) 

```
typedef struct test {  
    unsigned int a : 1;  
    unsigned int b : 1;  
    unsigned int c : 1;  
    unsigned int d : 1;  
    unsigned int e : 1;  
    unsigned int f : 1;  
    unsigned int g : 1;  
    unsigned int h : 1;  
    unsigned int i : 1;  
    unsigned int j : 1;  
    unsigned int k : 1;  
    unsigned int l : 1;  
    unsigned int m : 1;  
    unsigned int n : 1;  
    unsigned int o : 1;  
    unsigned int p : 1;  
}test;
```

---

7. #include <stdio.h>

```

typedef struct NODE {
    char value;
    struct NODE* next;
}node_t;

typedef struct LIST {
    node_t* head;
}list_t;

node_t* end(list_t list){
    node_t* current = list.head;
    while(current->next != NULL) { current = current->next; }
    return current;
}

void append(node_t* node, list_t list) {
    node_t* a = end(list);
    a->next = node;
}

// caution: destructive
void concatenate(list_t first, list_t second) {
    append(second.head, first);
    second.head = first.head;
}

void printCharList(list_t list) {
    node_t* current = list.head;
    while(current->next != NULL) {
        printf("%c", current->value);
        current = current->next;
    }
    printf("%c", current->value);
}

// test program that concatenates two char lists
int main() {
    node_t a = {'y', NULL};
    node_t b = {'o', NULL};
    list_t list1 = {&a}; // first list
    append(&b, list1);

    node_t c = {' ', NULL};

```

```

node_t d = {'m', NULL};
node_t e = {'a', NULL};
node_t f = {'n', NULL};
list_t list2 = {&c}; //second list
append(&d, list2);
append(&e, list2);
append(&f, list2);

concatenate(list1, list2);
printCharList(list1);
printf("\n");
return 0;
}

```

---

```

742 From https://github.com/dave-maldonado/COP3514
743     e032a1f..0447bee  master    -> origin/master
744 Updating e032a1f..0447bee
745 Fast-forward
746  hw3_ll_concat.c | 58 +++++
747  +++++
748  1 files changed, 58 insertions(+), 0 deletions(-)
749  create mode 100644 hw3_ll_concat.c
750 [dmaldonado1@c4lab02]~/COP3514% gcc -std=c99 -Wall -o hw3_ll_concat hw3_l
751 l_concat.c
752 [dmaldonado1@c4lab02]~/COP3514% ./hw3_ll_concat
753 yo man
754 [dmaldonado1@c4lab02]~/COP3514% █

```

-:\*\*\*- \*terminal<1>\* Bot L754 (Term: char run pair WS)

Figure 7: solution for problem 7 compiling and running

8. I didn't get this one working in time. I think my algorithm is ok but I have a bug somewhere.

```
_____ hw3_ll_reverse.c _____  
#include <stdio.h>  
#include <stdlib.h>  
  
typedef struct NODE {  
    char value;  
    struct NODE* next;  
}node_t;  
  
typedef struct LIST {  
    node_t* head;  
}list_t;  
  
node_t* end(list_t list){  
    node_t* current = list.head;  
    while(current->next != NULL) { current = current->next; }  
    return current;  
}  
  
void append(node_t* node, list_t list) {  
    node_t* a = end(list);  
    a->next = node;  
}  
  
list_t reverse(list_t list) {  
    node_t* current = list.head;  
    node_t* previous = NULL;  
    while(current) {  
        node_t* next = current->next;  
        current->next = previous;  
        previous = current;  
        current = next;  
    }  
    list.head = current;  
    return list;  
}  
  
void printCharList(list_t list) {  
    node_t* current = list.head;  
    while(current->next != NULL) {  
        printf("%c", current->value);  
        current = current->next;  
    }  
}
```

```

        printf("%c", current->value);
    }

    // test program that reverses char list
int main() {
    node_t a = {'d', NULL};
    node_t b = {'e', NULL};
    node_t c = {'c', NULL};
    node_t d = {'a', NULL};
    node_t e = {'l', NULL};
    list_t list = {&a};
    append(&b, list);
    append(&c, list);
    append(&d, list);
    append(&e, list);

    printCharList(list);
    printf("\n");
    list_t new_list = reverse(list);
    printCharList(new_list);
    printf("\n");

    return 0;
}

```

---

```

1021 Updating ee94d75..816a1e0
1022 Fast-forward
1023 hw3_ll_reverse | Bin 9200 -> 0 bytes
1024 hw3_ll_reverse.c | 68 ++++++++++++++++++++++++++++++++++++++
1025 ++++++
1026 2 files changed, 68 insertions(+), 0 deletions(-)
1027 delete mode 100755 hw3_ll_reverse
1028 create mode 100644 hw3_ll_reverse.c
1029 [dmaldonado1@c4lab02]~/COP3514% gcc -std=c99 -Wall -o hw3_ll_reverse hw3
1030 _ll_reverse.c
1031 [dmaldonado1@c4lab02]~/COP3514% ./hw3_ll_reverse
1032 decal
1033 zsh: segmentation fault (core dumped) ./hw3_ll_reverse
1034 [dmaldonado1@c4lab02]~/COP3514% 

```

Figure 8: solution for problem 8 compiling and not running :(