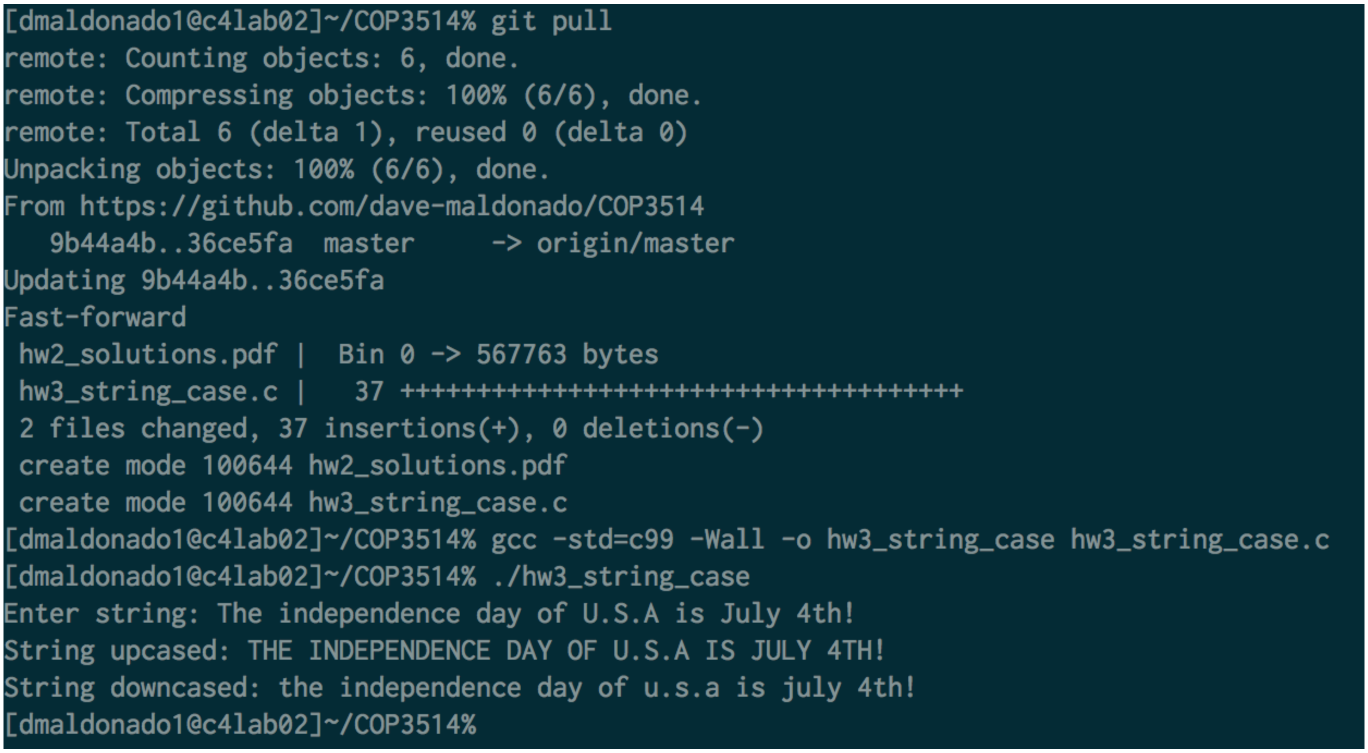
COP 3514 #3

David Maldonado

July 9, 2014

1. hw3_string_case.c

```c
#include <stdio.h>
#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;
}
```

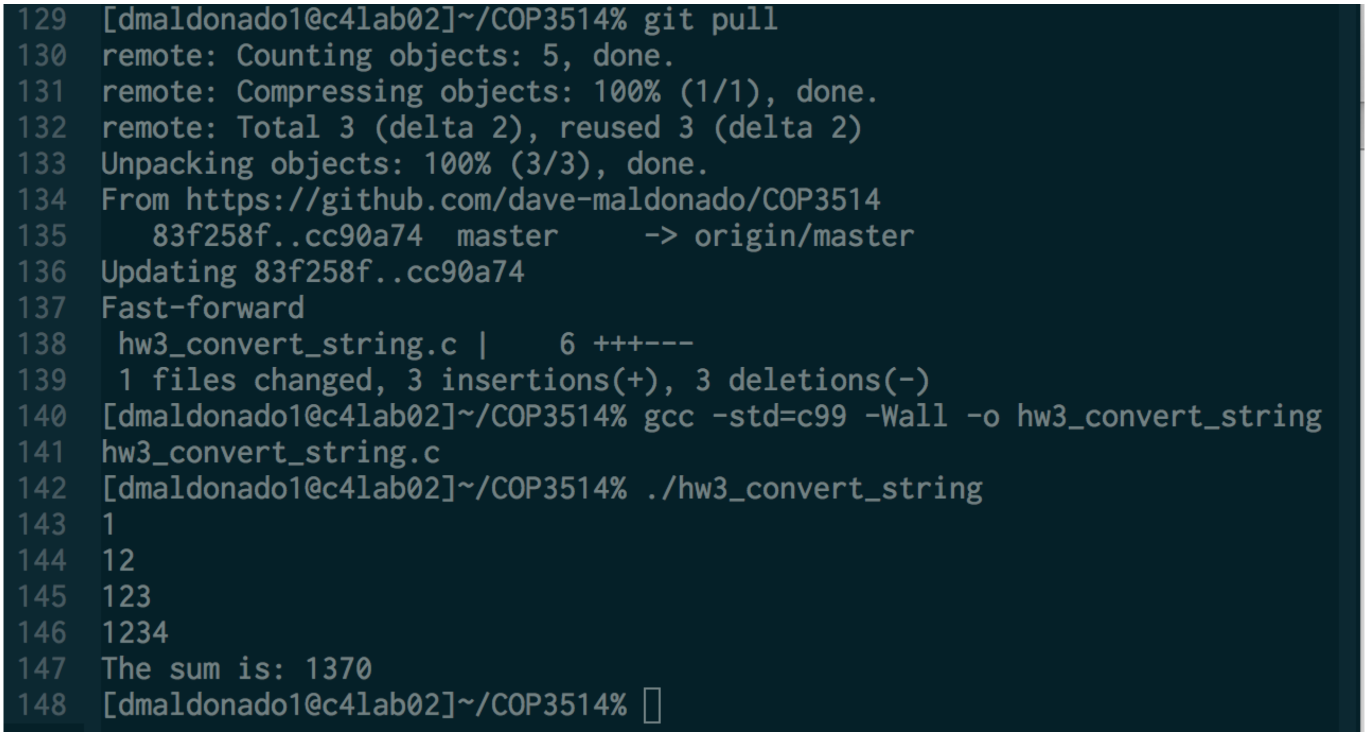


```
[dmaldonado1@c4lab02]~/COP3514% git pull
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 6 (delta 1), reused 0 (delta 0)
Unpacking objects: 100% (6/6), done.
From https://github.com/dave-maldonado/COP3514
   9b44a4b..36ce5fa  master     -> origin/master
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

hw3_convert_string.c

2.
```c
#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;
}
```



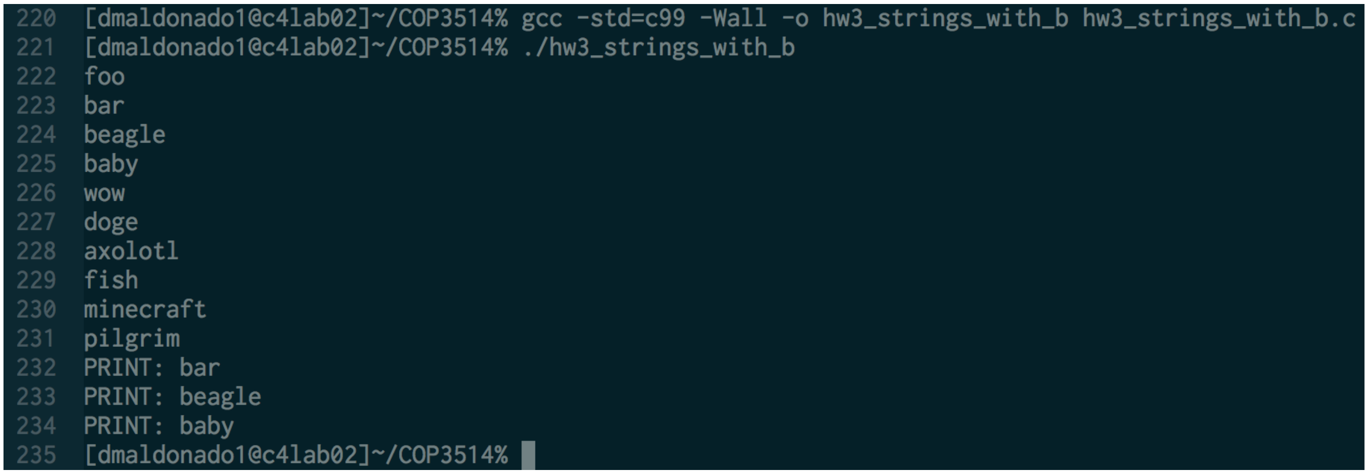
```
[dmaldonado1@c4lab02]~/COP3514% git pull
remote: Counting objects: 5, done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 3 (delta 2), reused 3 (delta 2)
Unpacking objects: 100% (3/3), done.
From https://github.com/dave-maldonado/COP3514
   83f258f..cc90a74  master     -> origin/master
Updating 83f258f..cc90a74
Fast-forward
 hw3_convert_string.c |    6 +++---
 1 files changed, 3 insertions(+), 3 deletions(-)
[dmaldonado1@c4lab02]~/COP3514% gcc -std=c99 -Wall -o hw3_convert_string
hw3_convert_string.c
[dmaldonado1@c4lab02]~/COP3514% ./hw3_convert_string
1
12
123
1234
The sum is: 1370
[dmaldonado1@c4lab02]~/COP3514% 
```

Figure 2: solution for problem 2 compiling and running

3. hw3_strings_with_b.c

```
#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++) {
```

```c
        if(startsWithb(strings[i])) { printf("PRINT: %s \n", strings[i]); }
    }
    return 0;
}
```



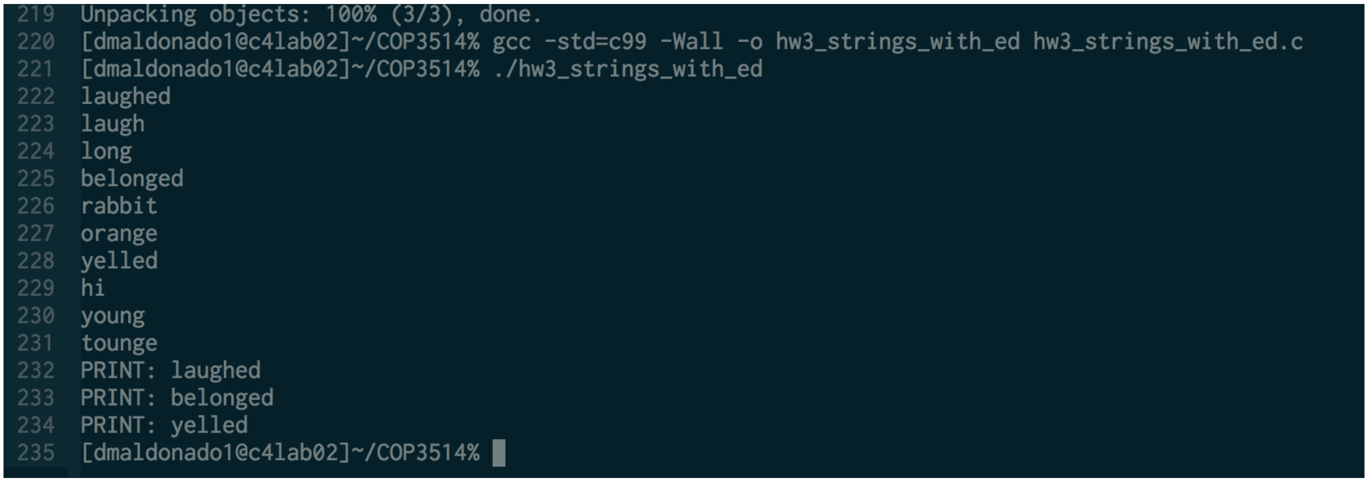
```
[dmaldonado1@c4lab02]~/COP3514% gcc -std=c99 -Wall -o hw3_strings_with_b hw3_strings_with_b.c
[dmaldonado1@c4lab02]~/COP3514% ./hw3_strings_with_b
foo
bar
beagle
baby
wow
doge
axolotl
fish
minecraft
pilgrim
PRINT: bar
PRINT: beagle
PRINT: baby
[dmaldonado1@c4lab02]~/COP3514%
```

Figure 3: solution for problem 3 compiling and running

hw2_strings_with_ed.c

4.

```c
#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;
}
```



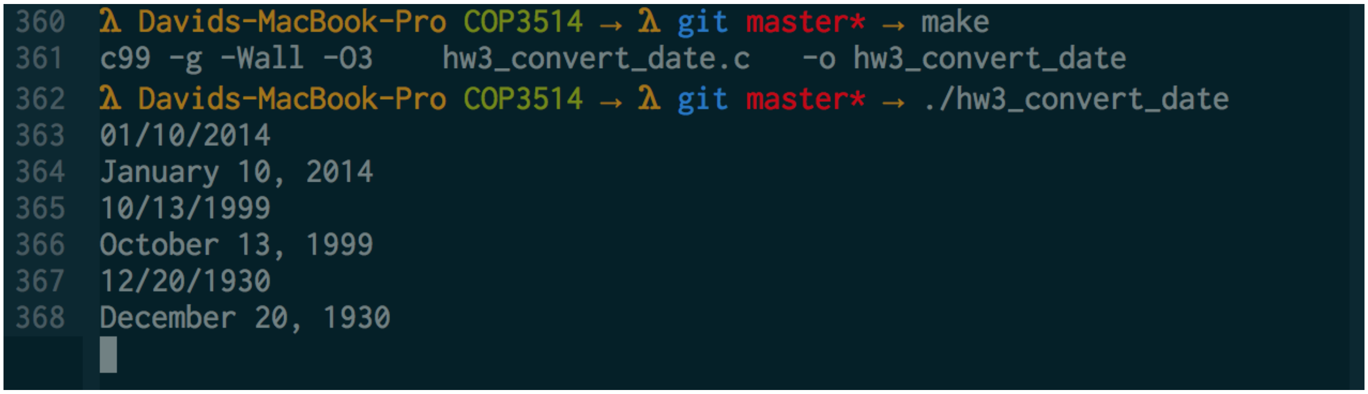
```
Unpacking objects: 100% (3/3), done.
[dmaldonado1@c4lab02]~/COP3514% gcc -std=c99 -Wall -o hw3_strings_with_ed hw3_strings_with_ed.c
[dmaldonado1@c4lab02]~/COP3514% ./hw3_strings_with_ed
laughed
laugh
long
belonged
rabbit
orange
yelled
hi
young
tounge
PRINT: laughed
PRINT: belonged
PRINT: yelled
[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.

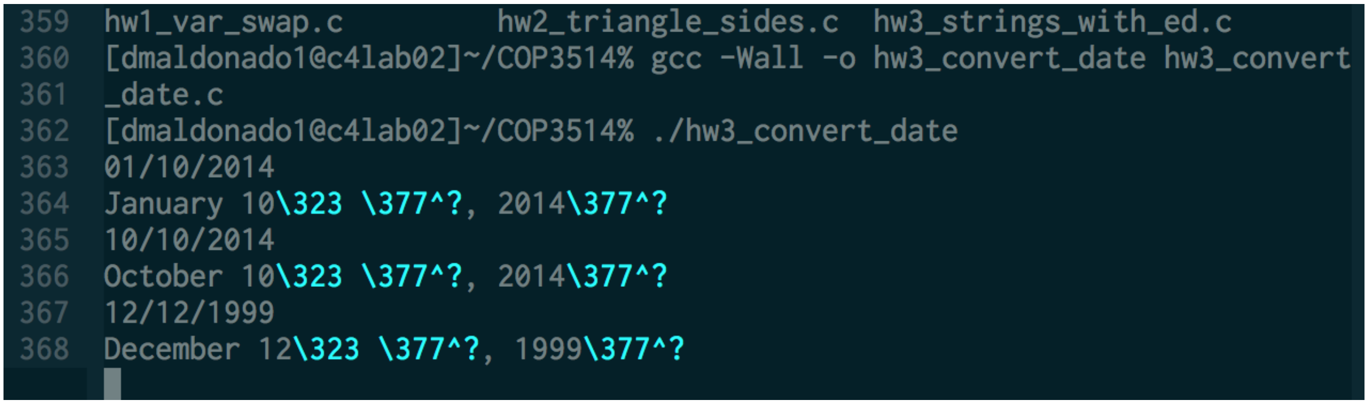
hw3_convert_date.c

```
#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;
}
```



```
λ Davids-MacBook-Pro COP3514 → λ git master* → make
c99 -g -Wall -O3     hw3_convert_date.c    -o hw3_convert_date
λ Davids-MacBook-Pro COP3514 → λ git master* → ./hw3_convert_date
01/10/2014
January 10, 2014
10/13/1999
October 13, 1999
12/20/1930
December 20, 1930
```

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



```
hw1_var_swap.c          hw2_triangle_sides.c  hw3_strings_with_ed.c
[dmaldonado1@c4lab02]~/COP3514% gcc -Wall -o hw3_convert_date hw3_convert
_date.c
[dmaldonado1@c4lab02]~/COP3514% ./hw3_convert_date
01/10/2014
January 10\323 \377^?, 2014\377^?
10/10/2014
October 10\323 \377^?, 2014\377^?
12/12/1999
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;
```

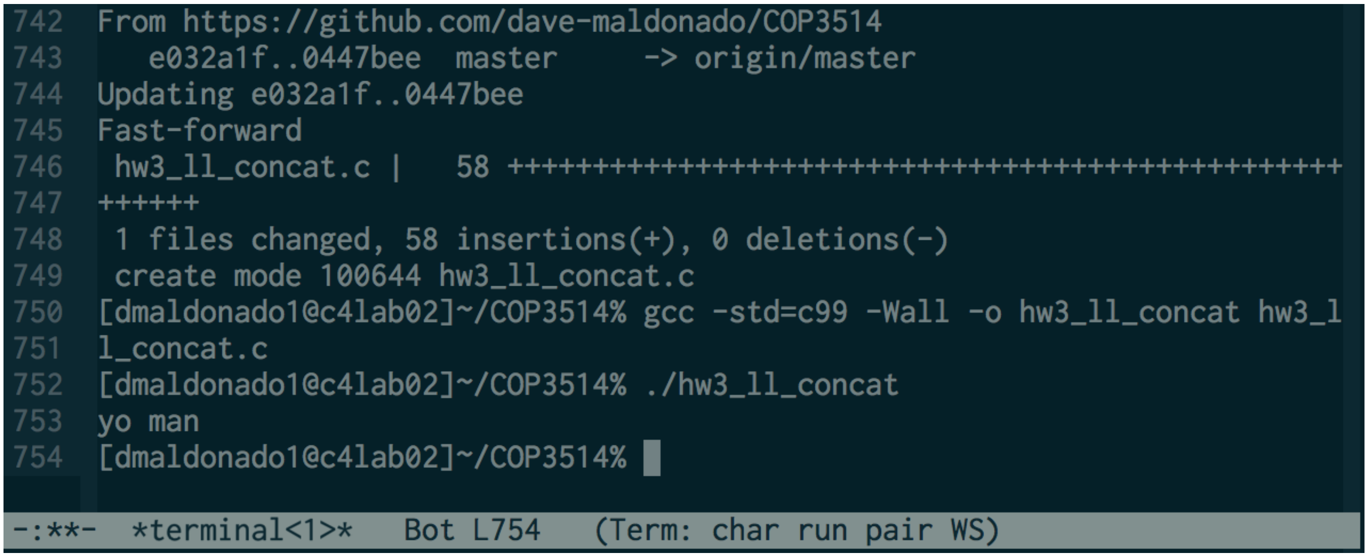
(e)

```
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;
```

hw3_ll_concat.c

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;
}
```



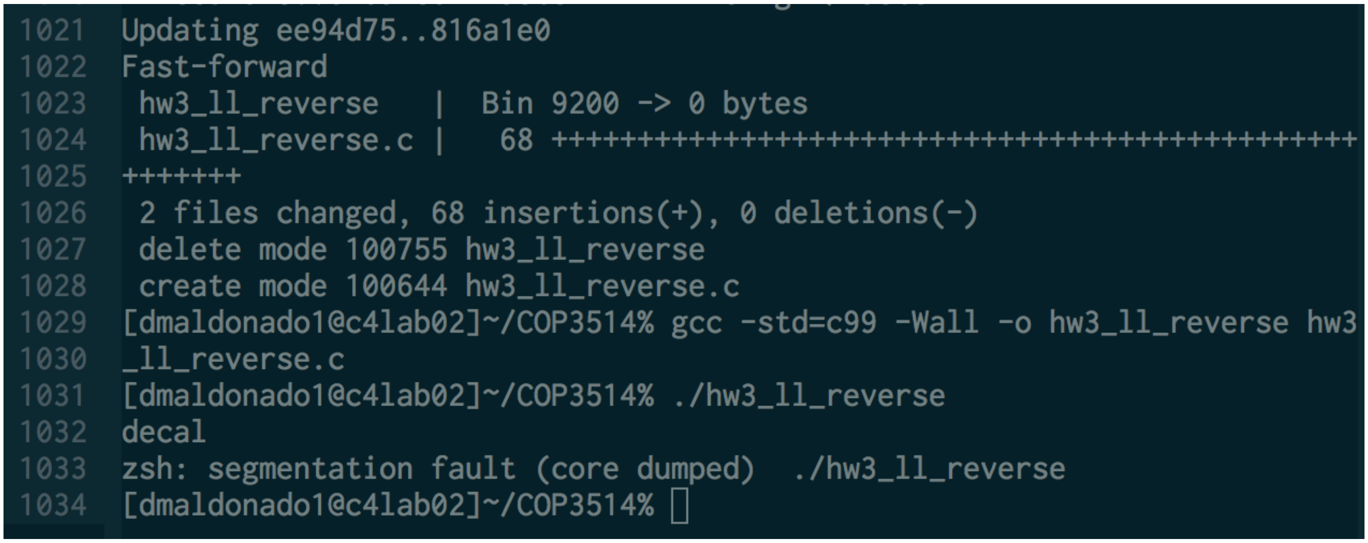
```
From https://github.com/dave-maldonado/COP3514
   e032a1f..0447bee  master     -> origin/master
Updating e032a1f..0447bee
Fast-forward
 hw3_ll_concat.c |   58 +++++++++++++++++++++++++++++++++++++++++++++++++
++++++
 1 files changed, 58 insertions(+), 0 deletions(-)
 create mode 100644 hw3_ll_concat.c
[dmaldonado1@c4lab02]~/COP3514% gcc -std=c99 -Wall -o hw3_ll_concat hw3_l
l_concat.c
[dmaldonado1@c4lab02]~/COP3514% ./hw3_ll_concat
yo man
[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_convert_date.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;
 }
```



```
Updating ee94d75..816a1e0
Fast-forward
 hw3_ll_reverse   |  Bin 9200 -> 0 bytes
 hw3_ll_reverse.c |   68 +++++++++++++++++++++++++++++++++++++++++++++++
+++++++
 2 files changed, 68 insertions(+), 0 deletions(-)
 delete mode 100755 hw3_ll_reverse
 create mode 100644 hw3_ll_reverse.c
[dmaldonado1@c4lab02]~/COP3514% gcc -std=c99 -Wall -o hw3_ll_reverse hw3
_ll_reverse.c
[dmaldonado1@c4lab02]~/COP3514% ./hw3_ll_reverse
decal
zsh: segmentation fault (core dumped)  ./hw3_ll_reverse
[dmaldonado1@c4lab02]~/COP3514% 
```

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