COP 3514 #2

David Maldonado

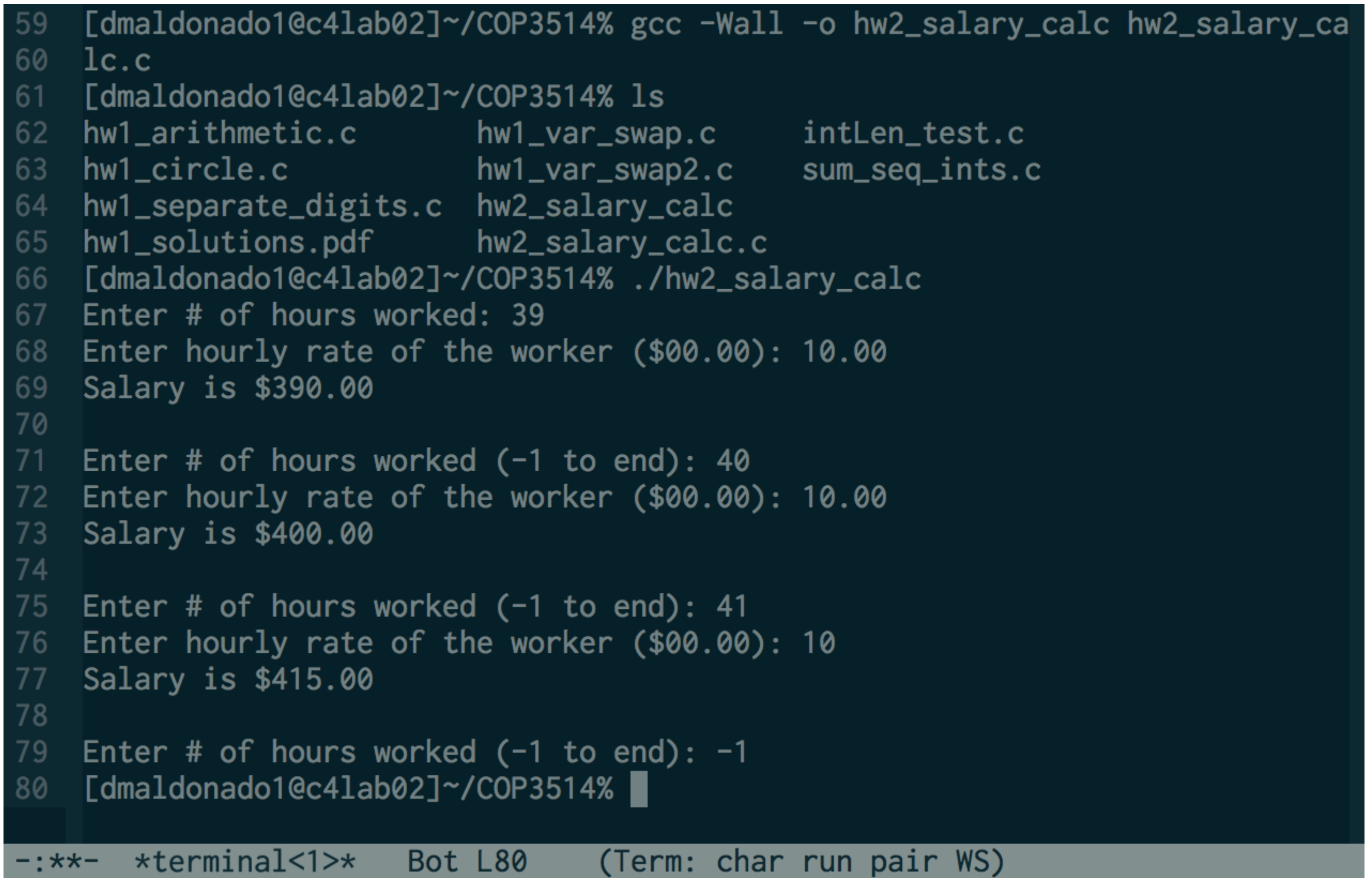
June 7, 2014

hw2_salary_calc.c

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

double calculateSalary(int hours, double rate) {
    int overtime = hours - 40;
    int salary;
    if (overtime > 0) {
        salary = 40 * rate + overtime * (rate + (rate / 2));
    } else {
        salary = hours * rate;
    }
    return salary;
}

// calculates salary given hours and hourly pay rate
int main() {
    int hours;
    double rate;
    printf("Enter # of hours worked: ");
    scanf("%d", &hours);
    while(hours != -1) {
        printf("Enter hourly rate of the worker ($00.00): ");
        scanf("%lf", &rate);
        printf("Salary is $%.2f \n\n", calculateSalary(hours, rate));
        printf("Enter # of hours worked (-1 to end): ");
        scanf("%d", &hours);
    }
    return 0;
}
```



```
[dmaldonado1@c4lab02]~/COP3514% gcc -Wall -o hw2_salary_calc hw2_salary_ca
lc.c
[dmaldonado1@c4lab02]~/COP3514% ls
hw1_arithmetic.c       hw1_var_swap.c     intLen_test.c
hw1_circle.c           hw1_var_swap2.c    sum_seq_ints.c
hw1_separate_digits.c  hw2_salary_calc
hw1_solutions.pdf      hw2_salary_calc.c
[dmaldonado1@c4lab02]~/COP3514% ./hw2_salary_calc
Enter # of hours worked: 39
Enter hourly rate of the worker ($00.00): 10.00
Salary is $390.00

Enter # of hours worked (-1 to end): 40
Enter hourly rate of the worker ($00.00): 10.00
Salary is $400.00

Enter # of hours worked (-1 to end): 41
Enter hourly rate of the worker ($00.00): 10
Salary is $415.00

Enter # of hours worked (-1 to end): -1
[dmaldonado1@c4lab02]~/COP3514% 

-:**-  *terminal<1>*   Bot L80    (Term: char run pair WS)
```

Figure 1: solution for problem 1 compiling and running

hw2_two_largest.c

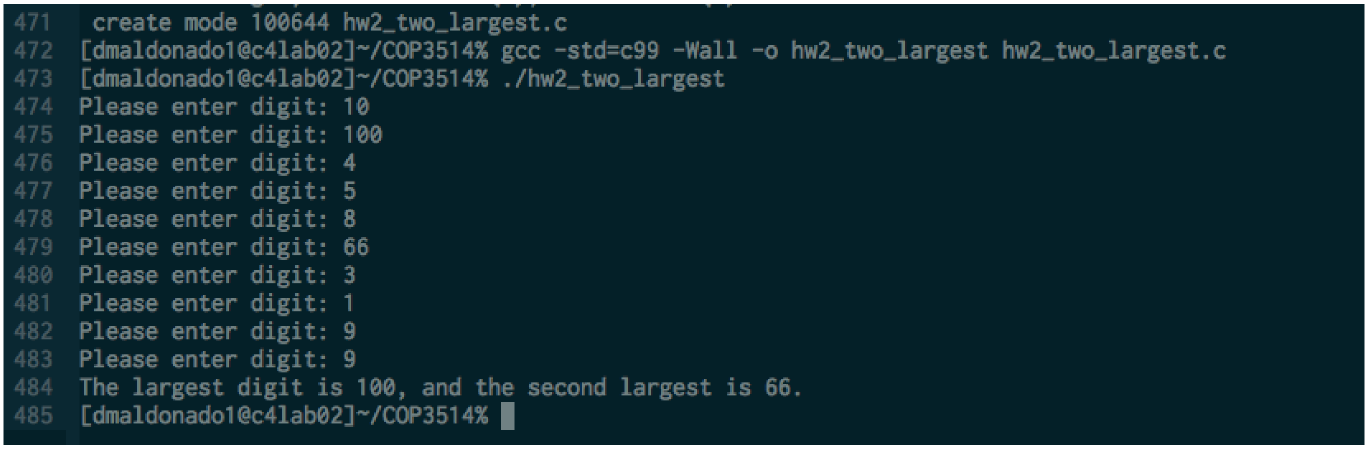
2.
```c
#include <stdio.h>

// return value of largest digit in array
int largest(int *digits, int n_digits) {
    int largest = digits[0];
    for(int i = 0; i < n_digits; i++) {
        if(digits[i] > largest) {
            largest = digits[i];
        }
    }
    return largest;
}

// sets largest digit(s) in array to 0
void clearLargest(int *digits, int n_digits, int largest) {
    for(int i = 0; i < n_digits; i++) {
        if(digits[i] == largest) { digits[i] = 0; }
    }
}

// find largest and second largest digits in array
int main() {
    int n_digits = 10;
    int digits[n_digits];
    int largest_digit, second_largest_digit;

    for(int i = 0; i < n_digits; i++) {
        printf("Please enter digit: ");
        scanf("%d", &digits[i]);
    }
    largest_digit = largest(&digits[0], n_digits);
    clearLargest(&digits[0], n_digits, largest_digit);
    second_largest_digit = largest(&digits[0], n_digits);
    printf("The largest digit is %d, ", largest_digit);
    printf("and the second largest is %d. \n", second_largest_digit);
    return 0;
}
```



```
 create mode 100644 hw2_two_largest.c
[dmaldonado1@c4lab02]~/COP3514% gcc -std=c99 -Wall -o hw2_two_largest hw2_two_largest.c
[dmaldonado1@c4lab02]~/COP3514% ./hw2_two_largest
Please enter digit: 10
Please enter digit: 100
Please enter digit: 4
Please enter digit: 5
Please enter digit: 8
Please enter digit: 66
Please enter digit: 3
Please enter digit: 1
Please enter digit: 9
Please enter digit: 9
The largest digit is 100, and the second largest is 66.
[dmaldonado1@c4lab02]~/COP3514%
```

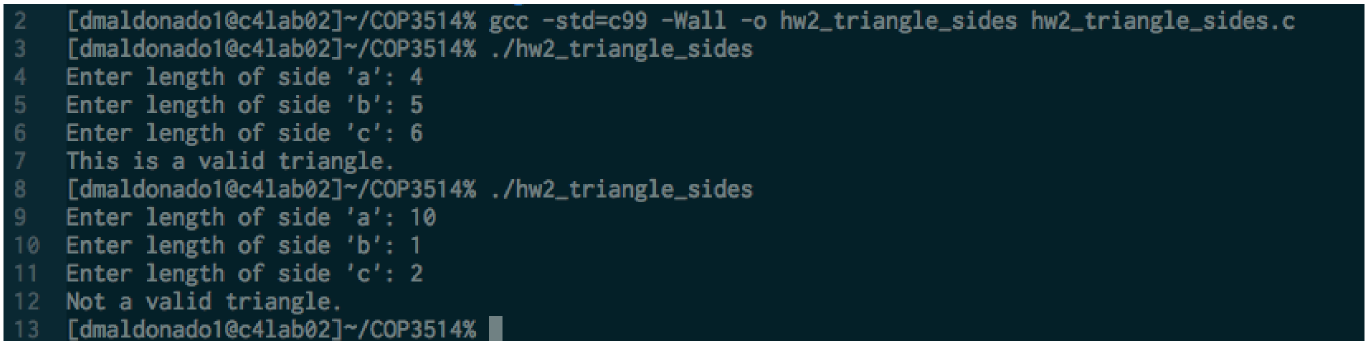
Figure 2: solution for problem 2 compiling and running

hw2_triangle_sides.c

3.
```c
#include <stdio.h>
#include <stdbool.h>

bool isTriangle(int a, int b, int c) {
    if((a < (c + b)) && (a > (c - b))) { return true; }
    else { return false; }
}

// determine if triangle is valid given sides
int main() {
    int a, b, c; //side lengths
    printf("Enter length of side 'a': ");
    scanf("%d", &a);
    printf("Enter length of side 'b': ");
    scanf("%d", &b);
    printf("Enter length of side 'c': ");
    scanf("%d", &c);
    if(isTriangle(a, b, c)) {
        printf("This is a valid triangle. \n");
    } else {
        printf("Not a valid triangle. \n");
    }
    return 0;
}
```



```
[dmaldonado1@c4lab02]~/COP3514% gcc -std=c99 -Wall -o hw2_triangle_sides hw2_triangle_sides.c
[dmaldonado1@c4lab02]~/COP3514% ./hw2_triangle_sides
Enter length of side 'a': 4
Enter length of side 'b': 5
Enter length of side 'c': 6
This is a valid triangle.
[dmaldonado1@c4lab02]~/COP3514% ./hw2_triangle_sides
Enter length of side 'a': 10
Enter length of side 'b': 1
Enter length of side 'c': 2
Not a valid triangle.
[dmaldonado1@c4lab02]~/COP3514%
```

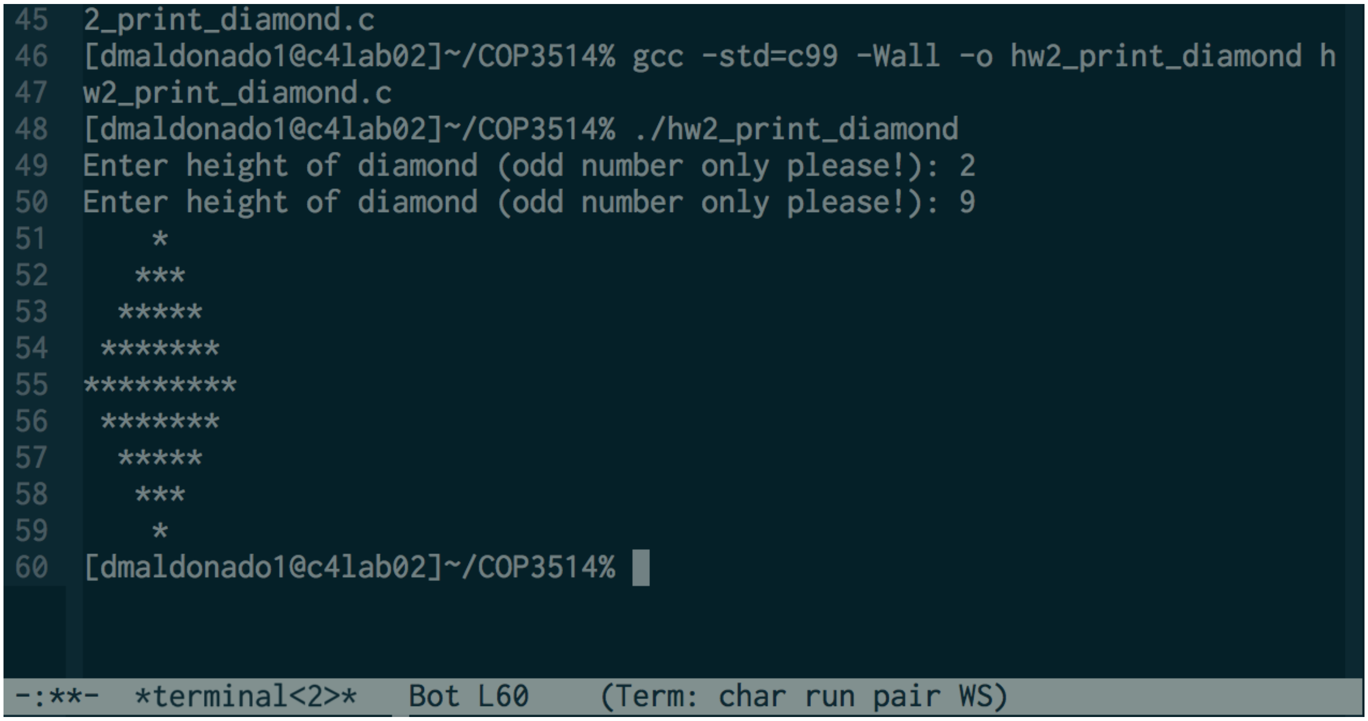
Figure 3: solution for problem 3 compiling and running

4. NOTE: Not satisfied with this solution, working on a refactor with an extracted makePyramid(int height, bool inverse) procedure.

hw2_print_diamond.c

```c
#include <stdio.h>

// given odd height prints asterisk 'diamond'
int main() {
    int height = 0;
    int spaces;
    int stars;
    while(height % 2 == 0) {
        printf("Enter height of diamond (odd number only please!): ");
        scanf("%d", &height);
    }
    spaces = (height - 1) / 2;
    stars = 1;
    // print top half
    while(spaces > 0) {
        for(int i = 0; i < spaces; i++) { printf(" "); }
        for(int i = 0; i < stars; i++) { printf("*"); }
        printf("\n");
        spaces--;
        stars += 2;
    }
    // print bottom half
    while(stars > 0) {
        for (int i = 0; i < spaces; i++) { printf(" "); }
        for (int i = 0; i < stars; i++) { printf("*"); }
        printf("\n");
        spaces++;
        stars -= 2;
    }
    return 0;
}
```



```
2_print_diamond.c
[dmaldonado1@c4lab02]~/COP3514% gcc -std=c99 -Wall -o hw2_print_diamond h
w2_print_diamond.c
[dmaldonado1@c4lab02]~/COP3514% ./hw2_print_diamond
Enter height of diamond (odd number only please!): 2
Enter height of diamond (odd number only please!): 9
    *
   ***
  *****
 *******
*********
 *******
  *****
   ***
    *
[dmaldonado1@c4lab02]~/COP3514% 

-:**-  *terminal<2>*   Bot L60     (Term: char run pair WS)
```

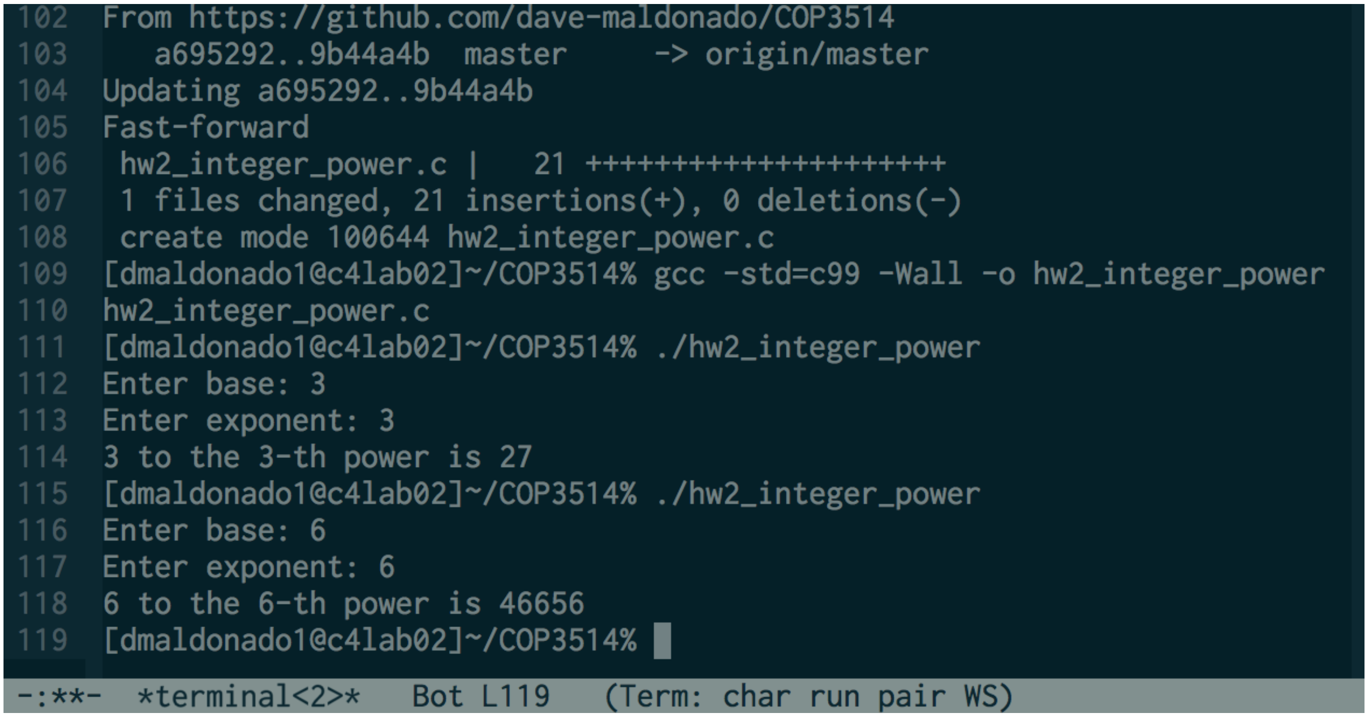
Figure 4: solution for problem 4 compiling and running

hw2_integer_power.c

5.
```c
#include <stdio.h>

int integerPower(int base, int exp) {
    int result = 1;
    if(exp == 0) { ; }
    else if(exp == 1){ result = base;}
    else { for(int i = 0; i < exp; i++) { result *= base; } }
    return result;
}

// prints base^exp
int main() {
    int base, exp;
    printf("Enter base: ");
    scanf("%d", &base);
    printf("Enter exponent: ");
    scanf("%d", &exp);
    printf("%d to the %d-th power is %d \n", base, exp, integerPower(base, exp));
    return 0;
}
```



```
From https://github.com/dave-maldonado/COP3514
   a695292..9b44a4b  master     -> origin/master
Updating a695292..9b44a4b
Fast-forward
 hw2_integer_power.c |   21 +++++++++++++++++++++
 1 files changed, 21 insertions(+), 0 deletions(-)
 create mode 100644 hw2_integer_power.c
[dmaldonado1@c4lab02]~/COP3514% gcc -std=c99 -Wall -o hw2_integer_power
hw2_integer_power.c
[dmaldonado1@c4lab02]~/COP3514% ./hw2_integer_power
Enter base: 3
Enter exponent: 3
3 to the 3-th power is 27
[dmaldonado1@c4lab02]~/COP3514% ./hw2_integer_power
Enter base: 6
Enter exponent: 6
6 to the 6-th power is 46656
[dmaldonado1@c4lab02]~/COP3514%
-:**-  *terminal<2>*   Bot L119    (Term: char run pair WS)
```

Figure 5: solution for problem 5 compiling and running