

CS16 Final Exam

E03, 10S, Phill Conrad, UC Santa Barbara

Wednesday, 06/09/2010

Name: _____

Umail Address: _____@ umail.ucsb.edu

Circle Lab section: 9AM 10AM 11AM Noon

[Link to Printer Friendly PDF Version](#)

Please write your name **only** on this page. That allows me to grade your exams without knowing whose exam I am grading.

This exam is **closed book, closed notes, closed mouth, cell phone off**, except for:

- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

There are 100 points worth of questions on the exam, and you have 3 hours (180 minutes) to complete the exam.

A hint for allocating your time:

- if a question is worth 10 points, spend no more than 10 minutes on it
- if a question is worth 20 points, spend no more than 20 minutes on it
- etc.

That will leave you with 80 extra minutes to check your work, or come to problems that gave you trouble.

1. (5 pts) **Read the instructions carefully and completely so you don't lose points!**

You were given an extra handout along with this exam, with a definition of a `struct Date`.

Using that definition, write a definition of `struct Hwk`—where `Hwk` stands for "Homework Assignment"—that contains the following fields.

For full credit, use **exactly the names and types given**.

- `num`, an integer representing the homework assignment number
- `earned`, an integer representing the points earned
- `poss`, an integer representing the points possible on that assignment.
- `date`, an instance of `struct Date` representing the date the student completed the assignment.

Material below this line goes with the question on the next page

```
int main()
{
    int failures=0;

    struct Hwk asns[]={    {1, 9, 10,{4,10,2010}},
                           {2, 9, 10,{4,10,2010}},
                           {3, 10, 10,{4,17,2010}} };

    // compute expected grade as a percentage
    double expected= ((9+9+10)*100.0)/(10+10+10);
    const double tol=0.0001;
    failures += CHECK_EXPECT_DOUBLE(overallGrade(asns,3),expected,tol);

    return tddFinish(failures);
}
```

2. (10 pts) Now, using the `struct Hwk` you defined in in the previous question, consider a function with the following prototype:

```
double overallGrade(struct Hwk *asns, int n);
```

The parameter `asns` represents an array of homework assignments (each one an instance of `struct Hwk`), and `n` represents the number of elements in that array. The function is intended to return the overall grade the student earned on all of the assignments, i.e. total points earned divided by total points possible, as a percentage.

The box on the previous page shows a test case for the function (written using the test-driven development framework we've been using throughout the course.)

The function definition below is correct, except that some lines of code are missing.

Fill in the missing lines of code so that the function works properly.

```
double overallGrade(struct Hwk *asns, int n)
{
    int totalPoss=0, totalEarned=0;

    int i;

    for (i=0; i<n; i++)
    {

    }

}
```

3. (10 pts) Consider the definition of `struct Circle` and `struct Point` on the handout that came with this exam.

Now consider this function prototype:

```
void moveCircle(struct Circle *c, double dx, double dy);
```

The purpose of this function is to move a circle by `dx` pixels in the x direction, and `dy` pixels in the y direction.

For example, here is a test case the function should pass (written using the test-driven development framework we've been using throughout the course):

```
int main()
{
    int failures=0;

    struct Circle c = { {10.0, 5.0}, 3.0} ;

    moveCircle(&c, 2.0, -3.0);
    failures += CHECK_EXPECT_INT(c.center.x, 10.0 + 2.0);
    failures += CHECK_EXPECT_INT(c.center.y, 5.0 - 3.0);

    return tddFinish(failures);
}
```

Write the **function definition** for `moveCircle`.

4. (18 pts) Give the value of each of these integer C expressions.

Express your answers in decimal (base 10).

Hint: remember the notations for hex and octal in C.

The first two are done for you as an examples.

Expression	Value in decimal
<code>0xA</code>	10
<code>1 ^ 0</code>	1
<code>4 ^ 2</code>	
<code>0xA & 0xC</code>	
<code>3 9</code>	
<code>7 && 4</code>	
<code>9 0</code>	
<code>1 << 4</code>	
<code>32 >> 1</code>	
<code>020</code>	
<code>0x20</code>	

5. (2 pts) As we discussed in class, the memory of a C program is divided into several parts: code, static area, stack, and heap.

Which part is the place local variables are stored?

Circle one: code static area stack heap

6. (2 pts) When you return from a function in C does the stack grow or shrink?

Circle one: grow shrink

7. Suppose I have the variables:

```
char thisCity[20] = "Santa Barbara";  
char thatCity[20] = "Los Angeles";
```

And then I have these lines of code:

```
thisCity[5]='\0'; // backslash zero  
thatCity[3]='0'; // just plain old zero
```

After executing these additional lines of code:

- a. (2 pts) What is the output of `printf("%s",thisCity);`

- b. (2 pts) What is the output of `printf("%s",thatCity);`

- c. (2 pts) What is the output of `printf("%i",strlen(thisCity));`

- d. (2 pts) What is the output of `printf("%i",strlen(thatCity));`

8. (10 pts) You are at a job interview for a job involving C programming. The interview seems to be going well, and then your potential new boss narrows her eyes, and gives you this funny look. She says:

So, last question. I hear that Conrad makes all his students memorize "an address is a pointer, and a pointer is an address".

*That's cute, and all, but I'm not convinced that his students understand what that **really means**. (I think they are just memorizing a saying.) So, explain it to me in a way that shows **you** understand, and you've got the job.*

How do you answer so that you get the job?

9. Here is the definition for a C function called `copyNonZero()` —with a few parts missing that you need to supply.

Fill in the missing lines of code in the places where you see comments with the `@@@` symbol

```
// Copy all numbers not equal to zero from array src into array dest.
// n is the occupancy of array src.
// Destination array should have capacity at least of size n.
// Return the number of values copied into dest
```

```
int copyNonZero(int *src, int *dest, int n)
{
    int i;
    int count=0;

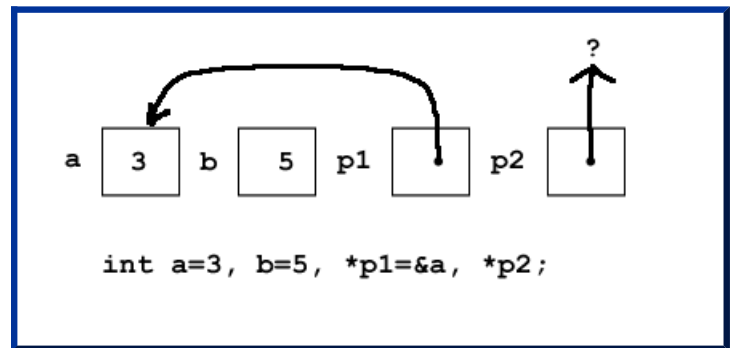
    for (i=0;i<n;i++)
    {
        if ( _____ != 0)    // @@@ (4 pts)

        {
            _____ = src[i]; // @@@ (4 pts)
            count++;
        }

    }

    return _____; // @@@ (2 pts)
}
```


10. Homework H15 involved an exercise that mimicked one from the Etter textbook, where you had to draw diagrams representing variable values and pointers. This exam question asks you to draw similar drawings (see box at right for an example.)



For each of the following, give the FINAL drawing—the one that would occur AFTER all the code executes. **Each problem starts from scratch.**

I've left space for "scratch work" as well as your final answer—I recommend you draw the initial state, make each change by crossing things out, then copy your final answer into the box given.

a. (5 pts)

<pre>int a=6, b=7, *p1=&b, *p2=&a; p1 = p2; (*p1) = 8; p2 = &b;</pre>	Scratch work
	Final answer

b. (5 pts)

<pre>int a=2, b=3, *p1, *p2; p2 = &a; p1 = &b; (*p1) = (*p1) + (*p2);</pre>	Scratch work
	Final answer

11. (10 pts) In project proj0, you were provided several definitions and function prototypes to work with. Here are a few of them:

```
struct Point {
    double x;

    double y;
};

void initPoint(struct Point *p, double xVal, double yVal);

struct Point makePoint(double xVal, double yVal);
void drawLine(struct Drawing *d, struct Point p1, struct Point p2, int color);
```

Using these, **write the function definition** for a function drawT that draws the letter T(as shown below).

Here is the function prototype for the drawT function you should define:

```
void drawT(struct Drawing *d,
           struct Point ul, // upper left corner
           double w, // width
           double h, // height
           int color);
```

And here is a sample main, and its output

```
int main()
{
    struct Drawing d;

    initDrawing(&d, DRAWINGTYPE_COLOR, 200, 100, COLOR_WHITE);

    drawT(&d,makePoint(10,10),20,50,COLOR_RED);
    drawT(&d,makePoint(100,10),50,20,COLOR_BLUE);

    writeFile(&d, FILENAME);

    return 0;
}
```



If you need extra room, use the space on the next page

Extra space for your drawT function (answer to question 11)

12. (5 pts) Here is a recursive function with the name `mysteryFunction`. As parameters, it takes an array of integers `a`, and the length of that array `n` as parameters, and returns an integer. What it actually computes is a mystery for you to solve.

```
int mysteryFunction(int *a, int n)
{
    if (n<2)
        return 1;
    if (a[0]>a[1])
        return 0;
    else
        return mysteryFunction(a+1, n-1);
}
```

Solve the mystery by figuring out what the mystery function computes, then decide which of the following is a more reasonable name for the function.

Among these possible function names, one (and only one) of them would be a reasonable answer to this question.

Just circle the correct answer.

average	indexFirstEven	maxValue
countEven	indexFirstOdd	minValue
countMax	indexLastEven	noDups
countMin	indexLastOdd	allDups
countNeg	indexOfMax	sum
countOdd	indexOfMin	sumEven
countPos	isSorted	sumNeg
countSevens	isAllEven	sumOdd
countZeros	isAllOdd	sumPos

End of Exam

Total Points: 100

CS16 Final Exam, Spring 2010

Extra Handout

Struct definitions for Questions 1 and 2:

```
struct Point {
    double x;
    double y;
};

struct Circle {
    struct Point center;
    double radius;
};

struct Date
{
    int m;
    int d;
    int y;
};
```