CS16 Final Exam
E03, 09F, Phill Conrad, UC Santa Barbara
Thursday, 12/10/2009

Name: ____

Umail Address: ______________________________@ umail.ucsb.edu

Circle Lab section: 8AM 10AM 11AM noon

Link to Printer Friendly PDF Version   Link to Sample Solutions

---

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) Write the definition of a C struct called Complex—a struct that represents a complex number such as (1.2 + 3.4i) with an "real part" and a "imaginary part". The struct should include two fields of type `double`: one called `a` for the real part, and another called b for the imaginary part.

2. Assume that you are continuing with the same program in which you defined `struct Complex` (from the previous question).
   a. (5 pts) Write the function prototype—for this quesiton **only the prototype**—for a `void` function called initComplex that takes three parameters:

      a. A parameter called `cPtr` that is a pointer to a `struct Complex`
      b. A parameter called `aVal` of type `double`
      c. A parameter called `bVal` of type `double`

   b. (10 pts) Now write the full function definition for initComplex. The function should initialize the values in the struct pointed to by cPtr with the values passed in in the two parameters, aVal and bVal.

3. (10 pts) First a math refresher:
   - Start with a complex number a + bi
   - If we take $(a+bi)^2$, we get $(a + bi)(a + bi) = a^2 + 2abi + b^2i^2$.
   - But since $i^2 = -1$, the result is really just $(a^2 - b^2 + 2abi)$.

   With this in mind, we can write a function that takes a struct Complex as a parameter, and returns a new struct Complex that is the square of the one passed in.

   Here is an incomplete definition for this function. Fill in the missing details.

   For full credit: do this by **using a function call to your initComplex function defined in the previous problem** to set the values in the variable `ans`.
   It is possible to write the answer to this problem with **just one line of code.**

   (Or, for partial credit, do it some other way.)

```
struct Complex squareComplex(struct Complex c)
{
  struct Complex ans;



  // You may declare additional variables and do
  // additional steps here if your answer requires them,
  // but there is a way to solve this that doesn't require that.






  // a function call to initComplex should come after this comment
  // that results in setting ans to the square of c








  return ans;
}
```

4. (5 pts) Suppose you are given a function definition in C.

   How can you tell whether that function is recursive or not—that is, what are you looking for?

   Be very precise in your use of terminology—be clear about the difference, for example, between function "definition", function "call", and function "prototype".

5. (15 pts) Write a C function definition for a function called `iStar` that takes two parameters, both integers, called `w` and `h` (standing for width and height), and returns an `int`.

   The purpose of the function is to print the letter I via `printf` calls, using stars (*), whenever the parameters for width and height have legal values.

   The parameters have legal values when the width is an odd number >= 3, and when height is >=3.

   When the parameters are NOT legal, your function should NOT print any error messages—in fact, it should not print ANYTHING. It should just return 1, indicating an error.

   Otherwise, print the I asked for, and return 0. Here is sample output for various function calls.

`iStar(3,4)`

```
***
 *
 *
***
```

`iStar(3,3)`

```
***
 *
***
```

`iStar(5,4)`

```
*****
  *
  *
*****
```

Assume that the following function is available to use, and make use of calls to it wherever possible.

```
void printNChars(int n, char c)

{

  int i;

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

    printf("%c",c);

}
```

Reminder: a **function definition**, NOT a main program!

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

6. Suppose you are the owner of a Unix file called `blah`.

   a. (2 pts) What Unix command could you have typed to see that the permissions on that file are, for example, `rw-r--r--`?

   b. (2 pts) What octal number corresponds to the permissions `rw-r--r--`?

   c. (2 pts) Suppose you want to type ./blah to execute the file blah as a program. Given the current permissions, that won't work. What is a unix command you can type to change the situation so that you can type ./blah to execute the program?

   d. (2 pts) What is the new permission string after you type that command?
   (Write your answer in the same format as rw-r--r--)

7. (15 pts) On lab08, 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);
```

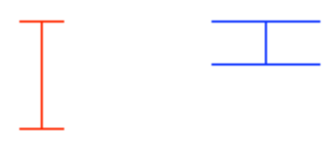
Using these, write the definition for a function drawI that draws the letter I (with bars on top and bottom, just like in the previous problem.)

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

```
void drawI(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);


  drawI(&d,makePoint(10,10),20,50,COLOR_RED);
  drawI(&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 answer to question 7

8. (12 pts) Together with this exam, there is a program (on a separate handout).

   Assuming each of the expressions below appeared in this program, indicate the type they would have, or write error if the expression is not valid, e.g.

   - dereferencing something with * or -> that isn't a pointer
   - a reference to a struct member that doesn't exist (e.g. d.foo where there is no foo)

   The first few are done for you as an example.

   Hints--for full credit:

   - don't write *pointer to character*; instead, write **char ***
   - don't write *address of int*; instead, write **int ***
   - don't write *address of int ** or *address of pointer to int*, instead write **int ****

   See solution

Expression	Type
a	double *
*b	error
*e	
&f	
*g	
f	
e->center.x	
&argc	

9. 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==0)
    return 0;

  if (a[0]%2!=0)
    return 1 + mysteryFunction(a+1, n-1);
  else
    return mysteryFunction(a+1, n-1);
}
```

a. (5 pts) Solve the mystery by figuring out what the mystery function computes, then come up with a more reasonable name for the function.

   Hint: Among these possible function names, one (and only one) of them would be a reasonable answer to this question: `average, countEven, countMax, countMin, countNeg, countOdd, countPos, countSevens, indexFirstEven, indexFirstOdd, indexOfMax, indexOfMin, isSorted, maxValue, minValue, noDups, sum`.

b. (10 pts) Rewrite the function so that it computes the same result for every input, but using iteration (a `for` loop or `while` loop) instead of recursion.

End of Exam

Total Points: 100

CS16 Final Exam
Extra Handout

Program for question about types

```
// types.c  Code for exam question, 11/15/2009
// P. Conrad for CS16, 09F, UCSB


#include <stdio.h>


struct Point {
  double x;
  double y;
};


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


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


int main(int argc, char *argv[])
{
  double *a;
  double b;
  int *c;
  int d;
  struct Circle *e;
  struct Circle f;
  struct Date *i;
  struct Date j;
  struct Point *g;
  struct Point h;

  // Program does no useful work
  // It is just the basis of a homework assignment about types



  // Pretend there is useful code here, and then
  // answer questions about the types of various expressions
  // as if they appeared right here.


  return 0;


}
```