CS16 Midterm Exam 2
E02, 09F, Phill Conrad, UC Santa Barbara
Wednesday, 11/18/2009

Name: ____

Umail Address: ______________________________________@ umail.ucsb.edu

Circle Lab section: 8AM 10AM 11AM noon

Link to Printer Friendly PDF Version Answer Key

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 50 minutes to complete the exam.

A hint for allocating your time:

- if a question is worth 10 points, spend no more than 5 minutes on it
- if a question is worth 20 points, spend no more than 10 minutes on it
- etc.

---

1. (20 pts) Write the definition of a C function initGPAreport according to the description in the comment below. Assume that the struct definition given is available to you.

   If you use any helper functions, then include definitions of those functions also.

   Write ONLY the function definition—for this question, I do NOT want a complete C program, so do NOT include any extraneous stuff such as #include <stdio.h> or a main function.

```
struct GPAreport {
  int perm; // perm number
  double gpa; // between 0.00 and 4.00
};


// FOR FULL CREDIT, FOLLOW THE INSTRUCTIONS EXACTLY, including
// using EXACTLY THESE VARIABLE NAMES for the parameters...

// The function initGPAreport should take these parameters:
//   gpaRpt: a pointer to a struct GPAreport called gpaRpt
//   thePerm:   a perm number (integer)
//   theGpa: a grade point average (number between 0.00 and 4.00)
// It should initialize the struct pointed to with the values passed in.
// It doesn't return anything.
```

2. (20 pts) Write the definition of the C function deansListCount according to the description in the comment below. Assume that the struct definition given is available to you.

   If you use any helper functions, then include definitions of those functions also.

   Write ONLY the function definition—for this question, I do NOT want a complete C program, so do NOT include any extraneous stuff such as #include <stdio.h> or a main function.Write the definition of a C function that takes two parameters—an array of integers, and a second integer indicating the size of the array.

```
struct GPAreport {
  int perm; // perm number
  double gpa; // between 0.00 and 4.00
};


// FOR FULL CREDIT, FOLLOW THE INSTRUCTIONS EXACTLY, including
// using EXACTLY THESE VARIABLE NAMES for the parameters...

// This function deansListCount should take two parameters:
//  (1) an array of GPAreports called gpas
//  (2) an integer called n, which indicates the size of the gpas array
// It should return the number of GPAs in the array that are 3.5 or higher
```

3. (40 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		Expression	Type
a	double *		i.x	
*b	error		i->y	
e			&(j.y)	
b			(*g).x	
&b			(*g)->y	
*a			h.x	
*d			*(h.x)	
e.center			&g	
e->x			argc	
f->y			argv[0]	
e->center->x			argv[0][0]	

4. As you know, the ls -l command will give you information about the permissions that apply to accessing files in a format such as `rwxr-xr-x`

   a. (2 pts) Suppose you use the command from question 7, and you see that the current permission mode for a file called `blah` is `rw-r--r--`.

      Does the owner of this file have permission to run this program as a command by typing `./blah` at the Unix prompt?

   b. (2 pts) If you want to change the permissions of a file, there is a unix command you can use to do this of the form *commandName octalNumber filename*.

      Give this command (the octal number form) for changing the permissions of the file `foobar` to have the permission string `r-xr-xr-x`

   c. (2 pts) If you use the same command as the previous problem but with the octal number 755, what would the resulting rwx format string look like?

5. (9 pts) Suppose that you are performing the selection sort algorithm—as explained in lecture, in our "field day" exercise, and in the code for lab06, i.e.

   - After the entire sort, elements will be in order from smallest to largest
   - Each pass through the array "selects" the largest element from the smaller array, and swaps it into place (sometimes "swapping with itself" if it is already in place)

   With that in mind, indicate the values of each element of the array after the 1st, 2nd and 3rd swap.

   Keeping in mind that some of these swaps may be cases of an element being swapped with itself *—those still "count" as swaps!*

	a[0]	a[1]	a[2]	a[3]	a[4]
initial values	42	88	13	9	27
after 1st swap					
after 2nd swap					
after 3rd swap					
after 4th swap (final array)	9	13	27	42	88

6. (5 pts) What is an example of a file that does NOT contain machine languages instructions but for which it is useful and reasonable to set execute permissions?

   Briefly explain your answer.

   Hint: recall your work from lab07a

End of Exam

Total Points: 100

CS16 Midterm Exam 2
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;


}
```