

# CS16 Midterm Exam 2

## E02, 10W, Phill Conrad, UC Santa Barbara

### Tuesday, 03/02/2010

Name: \_\_\_\_\_

Umail Address: \_\_\_\_\_@ umail.ucsb.edu

Circle Lab section:                      3PM                      4PM                      5PM

[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 75 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.

This will leave you 25 minutes to go back and check your paper, and/or finish anything you didn't finish the first time around.

---

1. (10 pts) Write the definition of a C function `initTsunamiRpt()` according to the description in the comment below.

Assume that the struct definition given is available to you.

**To get full credit:**

- FOLLOW THE INSTRUCTIONS IN THE COMMENT BELOW EXACTLY in terms of **choice of variable names**.
- 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 TsunamiRpt {  
    int hr; // hour according to 24 hour clock, 0=midnight, 23 = 11pm  
    int min; // minutes between 0-60  
    double waveHt; // wave height in meters  
};
```

```
// The function initTsunamiRpt should take these parameters:  
//   trPtr: a pointer to a struct Tsunami  
//   hour: the hour the Tsunami hit (an integer)  
//   minute: the minute the Tsunami hit (an integer)  
//   height: the weight of the Tsunami wave, in meters (a double)  
//  
// It should initialize the respective members of the struct  
//   with the values passed into the function  
// The function doesn't return anything.
```

2. Here is a portion of a main program that would use the `initTsunamiRpt()` function you defined in the previous problem.

This program takes the hour, minute, and wave height of a tsunami as command line parameters, initializes a struct `TsunamiRpt` called `tr`, and then prints out the values in that struct.

Your job is simple: **supply exactly two missing lines of code:**

- (5 pts) The **function prototype for `initTsunamiRpt()`**, at the place indicated below
- (5 pts) A **function call to `initTsunamiRpt()`**, as the place indicated below

```
// tsunamiRpt.c    Exam problem for CS16 Midterm 2, Winter 2010
// P. Conrad, CS Dept, UC Santa Barbara

#include <stdio.h>
#include <stdlib.h> // for atof

struct TsunamiRpt
{
    int hr;
    int min;
    double waveHt;
};

// (a) Add function prototype for initTsunamiRpt here


int main(int argc, char *argv[])
{
    // declare variables
    struct TsunamiRpt tr;
    int hour, min;
    double height;

    // check arguments
    if (argc!=4)
    {
        printf("Usage: ./tsunamiRpt hour min height\n"); return 1;
    }

    // convert cmd lind args
    hour=atoi(argv[1]); min=atoi(argv[2]); height=atof(argv[3]);

    // (b) call initTsunamiRpt to initialize the tr variable
    // passing in the values that were converted from the command line


    // In a real program, additional code would go here to use the
    // struct TsunamiRpt variable tr in some way---sending it over the internet,
    // adding it to a database, or doing some calculation on it, for example...

    printf("The tsunami arrived at %02i:%02i with height %lf meters\n",
           tr.hr, tr.min, tr.waveHt);

    return 0;
}

// The function definition from question 1 would go here,
// or it could be compiled in a separate file---you DON'T need to
// rewrite it here on this exam though!
```

3. (30 pts) For each of the conversions below, give the correct answer.

Convert	From	To	Your answer
010 110	Binary	Octal	
96	Base 10	Base 2	
75	Base 8	Binary	
1001 0101 0100 1010	Hex	Binary	
0100 0010	Binary	Decimal	
A2D2	Hex	Base 2	

4. (50 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 member `foo` inside `d`)

The first one is 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 \*\***

Expression	Type		Expression	Type
a	int		i.d	
*a			i->y	
*b			&(j->y)	
b			*(j->x)	
&b			(*j).x	
*c			(*j)->x	
&d			g->center	
e			g.center	
*f			g.center.x	
e->m			h->center	
&(f->d)			h->center.x	
(*f).y			h->center->y	
(*f)->y			&(h->center.x)	

---

# End of Exam

**Total Points: 100**

# CS16 Midterm Exam 2, Winter 2010

## 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[])
{
    int a;
    int *b;
    double c;
    double *d;
    struct Date e;
    struct Date *f;
    struct Point i;
    struct Point *j;
    struct Circle g;
    struct Circle *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;
}
```