

CS16 Midterm Exam 1

E01, 10S, Phill Conrad, UC Santa Barbara
Wednesday, 04/21/2010, 1pm-1:50pm

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 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.

You will then complete the exam in 50 minutes, and have 25 minutes remaining to check your answers, or go back and work on problems you were unable to complete the first time through.

1. (20 pts) In lab03, you were given an example of a function `starL` that produced ASCII art like this:

Function call	Output	Function call	Output	Function call	Output
<code>starL(3,4)</code>	<pre> * * * *** </pre>	<code>starL(3,5)</code>	<pre> * * * * *** </pre>	<code>starL(5,3)</code>	<pre> * * ***** </pre>

And you had an assignment to write a function `starT` that produced output like this:

Function call	Output	Function call	Output	Function call	Output
<code>starT(3,4)</code>	<pre> *** * * * </pre>	<code>starT(3,5)</code>	<pre> *** * * * * </pre>	<code>starT(5,3)</code>	<pre> ***** * * </pre>

Your job now is to write a function that produces output like this:

Function call	Output	Function call	Output	Function call	Output
<code>starC(3,4)</code>	<pre> *** * * *** </pre>	<code>starC(3,5)</code>	<pre> *** * * * *** </pre>	<code>starC(5,3)</code>	<pre> ***** * ***** </pre>

On the next page, you'll find the entire source code for the `starC.c` program, except for the body of the function definition of `starC`. Please fill in that body.

Be sure to read the comments (in **bold**) immediately before the function definition.

To help you, along with this exam, you should have recieved a handout with the entire source code of the `starL.c` program.

Put your answer on the next page

Your answer to the starC.c problem goes here—fill in the function body

```
// starC.c  A demonstration of ASCII Art printing C characters
// Exam question for CS16, Spring 2010, UCSB

#include <stdio.h>
#include <stdlib.h>
void starC(int width, int height); // function prototype

int main(int argc, char *argv[])
{
    int width, height;

    if (argc!=3)
    {
        printf("Usage: ./starC width height\n");
        return 1;
    }

    // get command line params
    width = atoi(argv[1]);
    height = atoi(argv[2]);

    starC(width,height);    // call the function

    return 0;
}

// A C function that prints the letter C with stars,
// at any width or height, provided both width and height are >= 3
// If either is less than 3, the function should print nothing.)
```

```
void starC(int width, int height)
{
```

```
    // your answer goes here
```

```
}
```

2. (15 pts) Write the definition of a C function `divSquares` that takes two parameters with the names *e* and *f*, both of type `double`.

The function should return the value of: $\frac{e^2}{f^2}$

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.

Be sure that you write your answer using only valid C—not in math notation.

The following information goes the question on the **next page (question 3)**

bold indicates what the user types; **grey** indicates the Unix prompt.
Everything else is output from the program.

Sample Output (printf and scanf method)	Sample Output (argc, argv method)
<pre>-bash-2.05b\$./calcSR Enter e: 2.0 Enter f: 4.0 Answer is 0.25 -bash-2.05b\$</pre>	<pre>-bash-2.05b\$./calcSR 2.0 4.0 Answer is 0.25 -bash-2.05b\$</pre>

Hints for full credit

- o Use correct % specifiers (%d or %i for int, and %lf for double)
- o Remember the special syntax detail that applies to using scanf.
- o Remember your #includes
- o Use "function prototypes" when function calls precede function definitions

3. (15 pts) (*Continued from previous question...*) Write a complete main program that calls the `divSquares` function you wrote the previous question, allowing the person running the program to specify values for e and f .

You have two choices:

- Use `printf` and `scanf` to ask the user to input values for e and f —**OR**
- Get the values of e and f from the command line (using `argc, argv`)

Either method is ok, but choose one and stick with it—don't mix the two.

Your program should then pass the values of e and f to a function call to `divSquares`. Store the value returned in a variable of type `double` called `answer`. Then print the value of `answer` with this line of code:

```
printf("Answer is %lf\n", answer);
```

See handout that comes with this exam for sample output and additional hints.

You do not need to recopy your entire function definition from the previous question—pretend it appears in the box at the bottom. Note this means you'll need a function prototype, since the main where your function call will appear comes before the function definition.

```
// function definition for divSquares
```

Pretend your function definition for `divSquares` from the previous problem appears right here. You don't need to recopy it.

4. (16 pts) For each of the following C expressions, fill in the value, and then circle the type. The first two are done for you as an example.

Don't forget to fill in BOTH the value AND circle the type.

expression	value	type
3	3	<u>int</u> double char char *
0.5 * 0.5	0.25	int <u>double</u> char char *
19 % 4		int double char char *
37 % 100		int double char char *
3 + 7 * 2		int double char char *
1 + 1 / 5.0		int double char char *
'3'		int double char char *
5/4		int double char char *
4/5		int double char char *
"5 % 2"		int double char char *

5. (3 pts) Convert from binary to octal:

100 001 110

6. (3 pts) Convert from decimal to binary

62

7. (3 pts) Convert from base 16 to base 2

af72

8. (16 pts) For each of the for loops below:

a. Circle **infinite** if it is an infinite loop, or **finite** if it NOT an infinite loop

b. Check the in the **no output column** (☒) if the loop has no output

c. If the loop has output, put it in the box.

Note: if the output will be infinite, just write the **output of the first three times through the loop**, then put three dots like this: ...

code	Infinite or finite?	No output?	write the output here (if any)
<pre>int i; for (i=4; i>1; i--) printf("%d ",i);</pre>	infinite finite	<input type="checkbox"/>	
<pre>int i; for (i=1; i<=4; i++) printf("%d ",i);</pre>	infinite finite	<input type="checkbox"/>	
<pre>int i; for (i=1; i>=4; i++) printf("%d ",i);</pre>	infinite finite	<input type="checkbox"/>	
<pre>int i; for (i=0; i<4; i--) printf("%d ",i);</pre>	infinite finite	<input type="checkbox"/>	

9. Suppose we have a program where the main starts with the line:

```
int main(int argc, char *argv[])
```

and the program is run with the following command line:

```
./myprog 4 ever
```

a. (3 pts) What is the value of `argc`?

b. (3 pts) What is the value of `argv[2][2]`?

10. (3 pts) Suppose you have a different program that starts like this:

```
// reserveIt.c Seat reservation system, P. Conrad, 4/21/2010

#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    int numSeats;

    if (argc != 2)
    {
        printf("Usage: ./reserveIt numSeats\n");
        return 1;
    }

    // now assign the variable numSeats from the command line
```

Given the comment, and what you know about how command line arguments work, what should the next line of code be?

End of Exam

Total Points: 100

CS16 Midterm Exam 1

E01, 10S, Phill Conrad, UC Santa Barbara

Handout with starL.c

```
// starL.c  A demonstration of ASCII Art printing L characters
// P. Conrad for CS16, Winter 2010, UCSB

#include <stdio.h>
#include <stdlib.h>

void starL(int width, int height);

int main(int argc, char *argv[])
{
    int width, height;

    if (argc!=3)
    {
        printf("Usage: ./starL width height\n");
        return 1;
    }

    // remember argv[0] is the name of the program
    width = atoi(argv[1]);
    height = atoi(argv[2]);

    // call the function
    starL(width,height);

    return 0;
}

// A C function that prints the letter L with stars,
// at any width or height, provided both width and height are >= 2
// If either is less than 2, the function should print nothing.)

void starL(int width, int height)
{
    int i;

    // check if parameters are valid
    if ((width<2) || (height < 2))
        return; // return without printing anything

    // print height-1 rows of *\n
    for (i=0; i<height-1; i++)
        printf("*\n");

    // print width stars, followed by a final \n
    for (i=0; i<width; i++)
        printf("*");

    printf("\n");

    return; // we are finished
}
```