CS16 Midterm Exam 1

E01, 10W, Phill Conrad, UC Santa Barbara

Tuesday, 02/02/2010, 11am-12:15pm

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.

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. (10 pts) Write the definition of a C function that takes two integers as parameters, a and b. The function should return the value of the square of the difference between the two numbers, i.e. $(a-b)^2$.

   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.

2. (20 pts) There is an incomplete program below. It includes: the skelton of a `main()` function (incomplete), and function definitions for

   - `nStars`, which prints n '*' characters and
   - `starBox` makes prints a box of stars with "ASCII Art".

   Add the following code to make this an interactive program:

   - Add code in main that prompts the user for input (using printf/scanf), asking for both width and height.
   - Add code that calls starBox to output the box of stars.
   - Add anything else that is needed above the main() function so that the program will compile.

Sample Output

```
-bash-2.05b$ ./starBox
Please enter width: 4
Please enter height: 3
****
****
****
-bash-2.05b$
```

Hints for full credit

- Use correct % specifiers (%d or %i for int, and %lf for double)
- Remember the special syntax detail that applies to using scanf.
- Remember the line of code to access the standard input/output library
- Use "function prototypes" when function calls precede function definitions
- Do what is needed at the end of the output
  so that the Unix prompt after the program ends appears on a new line
- When the main() is finished, remember to signal success/failure to the OS

```
// starBox1.c   Answer to an exam question
// P. Conrad, for CS16   02/02/2010




int main()
{











}





void nStars(int n)
{
  int i;
  for (i=0; i<n; i++)
      printf("*");

}

void starBox(int width, int height)
{
   int i;
   for (i=0; i<height; i++)
   {
       nStars(width); printf("\n");
   }

}
```

3. (20 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.
(See solution)

expression	value	type
3	3	[✓ int] double char char *
0.5 * 0.5	0.25	int [✓ double] char char *
16 % 3		int double char char *
3 % 16		int double char char *
3 + 5 * 2		int double char char *
1 + 1 / 4.0		int double char char *
3.2/10		int double char char *
6 % 2		int double char char *
"3 + 4.0"		int double char char *
'3'		int double char char *
3 / 4		int double char char *
16 % 5		int double char char *

4. (10 pts) You are on a job interview with Jane Trotter, CEO of FlapOleo, a local software company. She starts to give you a quick tour of the company.

   As you enter the room where the programmers are working, she's telling you about how FlapOleo practices Test Driven Development. You nod and say that you've learned all about "TDD" in your CS classes at UCSB. Jane seems pleased.

   Then she points to a poster on the wall. It says:

   First, see the test fail
   Then, see the test pass

   Jane points out the poster. She says:

   Yesterday's job candidate didn't like this poster—saw it, and said that "real progammers write code that passes the test on the first try".

   That candidate was, indeed,a really good programmer according to the references we checked. But we didn't hire that person. Why do think we didn't?

   You can tell by the look in her eye that Jane is clearly testing you—she wants to see if you really understand test-driven development.

   Using what you learned in CS16 (in lab 2 and in lecture), what answer can you give Jane that will get you the job?

   (Note: Jane is also known to dislike long explanations, or wordy answers—so keep your answer short, but make sure it addresses the main point.)

5. (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)**

`int i;`
`for (i=0; i<4; i++)`
`  printf("%d ",i);`
	infinite finite	□	

`int i;`
`for (i=4; i>1; i--)`
`  printf("%d ",i);`
	infinite finite	□	

`int i;`
`for (i=1; i<3; i--)`
`  printf("%d ",i);`
	infinite finite	□	

`int i;`
 	
`for (i=1; i>3; i++)`
`  printf("%d ",i);`
	infinite finite	□	

6. (5 pts) Very briefly: What is the purpose of using int main(int argc, char *argv[]) instead of int main() in a C program?

7. In programs using `int main(int argc, char *argv[])` we often see lines of code such as the following:

```
width = atoi(argv[1]);
height = atoi(argv[2]);
```

Before we can use these lines of code, there are three things we must do.

   a. (3 pts) There is a line of code we need before the main program to be sure we can access `atoi`.

      What is that line of code?

   b. (3 pts) We need either one, or two lines of code we need right at the start of the main program to *declare* `width` and `height`.

      What is that code?

   c. (3 pts) There is one more thing we must do somewhere before the two calls to the atoi function above—we need to check if that something is true, and if so, we have to print an error message and stop the program.

      What is it that we have to check?

8. (10 pts) Consider the ASCII Art functions from lab03, lab04, and recent lectures.

In lecture, we covered a function that produced output like the following.

triangle1(0);	**triangle1(1);**	**triangle1(2;)**	**triangle1(3);**	**triangle1(4);**	**triangle1(5);**

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

Suppose that instead, we want a function that produces output like this:

triangle2(0);	**triangle2(1);**	**triangle2(2;)**	**triangle2(3);**	**triangle2(4);**	**triangle2(5);**

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

On the next page, you'll find the entire source code for the triangle2.c program, except for the body of the function `triangle2()`
Your job: fill in the body of that function.

To help you, along with this exam, you should have recieved a handout with the entire source code of the `triangle1.c` program.
You may use this as a reference.

Put your answer on the next page

Your answer to the triangle2.c problem goes here—fill in the function body

```
// triangle2.cc  A triangle of stars (upper-right corner)
// P. Conrad for CS16, Winter 2010

#include <stdio.h>
#include <stdlib.h>
// function prototypes
void printNChars(int n, char c);
void printKSpacesNStars(int k, int n);
void triangle2(int width);

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

  // check the number of arguments
  if (argc!=2)
    {
      printf("Usage: %s width\n", argv[0]);
      return 1;
    }

  // get the command line parameter
  width=atoi(argv[1]);
  triangle2(width); // draw the triangle

  return 0;
}

void triangle2(int width)
{
  int i;
  int height = width; // width is equal to height


   // your answer goes here

















}

void printNChars(int n, char c)
{
  int i;
  for (i=0; i<n; i++)
    printf("%c", c);
}

void printKSpacesNStars(int k, int n)
{
  printNChars(k,' ');  printNChars(n,'*');
}
```

End of Exam

Total Points: 100

CS16 Midterm Exam 1
E01, 10W, Phill Conrad, UC Santa Barbara
Tuesday, 02/02/2010, 11am-12:15pm

Handout with triangle1.c

```
// triangle1.cc  A triangle of stars (lower-right corner)
// P. Conrad for CS16, Winter 2010

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

// function prototypes
void printNChars(int n, char c);
void printKSpacesNStars(int k, int n);
void triangle1(int width);

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

  // check the number of arguments
  if (argc!=2)
    {
      printf("Usage: %s width\n", argv[0]);
      return 1;
    }

  // get the command line parameter
  width=atoi(argv[1]);
  triangle1(width); //function call to draw the triangle

  return 0;
}

void triangle1(int width)
{
  int i;
  int height = width; // width is equal to height

  // so this for loop iterates over the rows
  for (i=1;i<=height;i++)
    {
      // print width-1 spaces then a i stars and a newline
      printKSpacesNStars(width-i,i); printf("\n");
    }
}

void printNChars(int n, char c)
{
  int i;
  for (i=0; i<n; i++)
    printf("%c", c);
}

void printKSpacesNStars(int k, int n)
{
  printNChars(k,' ');  printNChars(n,'*');
}
```