

**NCSU ECE 209 Sections 602 and 604**  
**UNCA CSCI 373 Section 002**  
**Exam 1 Fall 2010**  
**5 October, 2010**

This is a closed book and closed notes exam. It is to be turned in by 12:05 pm. Calculators, PDA's, cell phones, and any other electronic or communication devices may not be used during this exam.

Please read and sign the following statement:

I have neither given nor received unauthorized assistance on this test.

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

If you want partial credit for imperfect answers, explain the reason for your answer!

**Problem 1 (5 points)**

Complete the following program so that it writes your name 209 times. Your name should be written on a separate line each of these 209 times.

```
#include <stdio.h>
#include <stdlib.h>
int main(void) {

    return(EXIT_SUCCESS) ;
}
```

**Problem 2 (9 points)**

In the nine boxes below are nine possible names for C identifiers. Cross out the ones that would **not** be legal C identifier names.

ten50	c#	_ten50
1050	int	Ten50
_ten_fifty_	©2010	ten50

**Problem 3 (6 points)**

In the six boxes below are six possible values for C integers. Cross out the ones that would **not** be legal literals for integer values in C.

107	3e3	x15
'A'	007	0x15

**Problem 4 (25 points)**

Below is a sequence of C statements defining some variables

```
int    xxi = 21 ;
char   chr = 'Q' ; /* ASCII code for 'Q' is 81 */
double xhf = 10.5 ;
```

The table below contains two columns. The leftmost column is a C expression. In the rightmost column write the value of the expression as a C literal. If the value is a double, be sure to use the C syntax for writing doubles. Some of these are tricky, but none require complex arithmetic.

10 + 5 * 4	
10 * 5 + 4	
10 + (5 * 4)	
10 * (5 + 4)	
5 % 3	
1 * 5.0 / 2	
xxi + 5 / 2	
xxi + 5.0 / 2	
(int) xxi + 2.5	
1 && 0	
1    0	
! 1	
9 / 4 * 3	
9 * 4 / 3	
9 / 4.0 * 3	
9 * 4 / 3.0	
5 > 5	
5 > 4 + 1	
5 > 4.0 + 1	
1 < 2 ? 4 : 6	
0 < 0 <= 0	
xxi = 5	
chr = xhf	
++chr	
chr++	

**Problem 5 (15 points)**

Each of the following four `for` or `while` loops, which are sometime preceded by a few initialization statements, print numbers. For each loop write in the four boxes below the loop the first four lines printed by loop. If the loop prints less than four lines, fill in a box for each line that is printed. You should assume that `i` has already been declared as an `int` variable.

```
for (i=4; i>=2; --i) {  
    printf("%d\n", i*i) ;  
}
```


```
i=0 ;  
while (i<4) {  
    printf("%d\n", ++i) ;  
}
```


```
for (i=0; i<=7; i=i+i) {  
    printf("%d\n", i+2) ;  
}
```


```
for (i=4; i; i = i-2) {  
    printf("%d\n", i) ;  
}
```


**Problem 6 (10 points)**

Suppose variables `c`, `d`, and `i` have been declared as follows:

```
char   c ;
double d ;
int    i ;
```

In the table below there is a `scanf` call in each cell of the first column. The remaining columns are labeled `c`, `d`, and `i`. Assume that **all** of the `scanf` calls are made with the input line shown below, which you has no whitespace at the beginning or end of the line and has single blanks separating the words and numbers.

```
3.1e4 2010 x
```

For each row, in the cells of the columns labeled `c`, `d`, and `i`, write the values of those three variables after the `scanf` in the leftmost cell of the row is executed. If the `scanf` does not change a variable, simply write a line across the cell for that variable.

The first two rows of the table have been filled in as examples.

	<code>c</code>	<code>d</code>	<code>i</code>
<code>scanf("%lf %d %c", &amp;d, &amp;i, &amp;c)</code>	'x'	31000.0	2010
<code>scanf("%c", &amp;c)</code>	'3'	_____	_____
<code>scanf("%3lf%c%d", &amp;d, &amp;c, &amp;i)</code>			
<code>scanf("%lf %c%o", &amp;d, &amp;c, &amp;i)</code>			
<code>scanf("%d%lf", &amp;i, &amp;d)</code>			
<code>scanf("%lf 20%c", &amp;d, &amp;c)</code>			

**Problem 7 (10 points)**

Given an `int` variable `n` that has already been declared and initialized to a positive value, and another `int` variable `j` that has already been declared, use a `while` loop to print a single line consisting of the letters “help” followed by `n` exclamation points. For example if `n` contains 5, then `help!!!!` would be printed.

**Problem 8 (5 points)**

Write a little section of code that uses an `int` variable `n` (which has already declared) and prints “unacceptable triangle size” if `n` is even or negative or greater than 25. (This should be very familiar.)

**Problem 9 (10 points)**

Suppose there is a `double` variable `E` with the value `2.718` and an `int` variable `Q` with the value `81`. Your task for this problem is to fill in the format string for the `printf` in the right column, so it will output **exactly** the line shown in the left column. Your answer must also actually use all the arguments passed to `printf`.

Because blanks are so important in the target output, a shadowy tilde `~` is being used to indicate blanks. But sure to do something similar in your answer.

The first row has been completed as an example. By the way, `0x51`, `0121`, and `'Q'` are other ways of saying `81` in C.

<code>2.718~~~81</code>	<code>printf("%g%5d\n", E, Q) ;</code>
<code>(2.718,81)</code>	<code>printf("          \n", E, Q) ;</code>
<code>~~2.7180~~Q</code>	<code>printf("          \n", E, Q) ;</code>
<code>~2.72e+00~~51</code>	<code>printf("          \n", E, Q) ;</code>
<code>2.781</code>	<code>printf("          \n", E, Q) ;</code>
<code>~~~3~~0121</code>	<code>printf("          \n", E, Q) ;</code>

**Problem 10 (5 points)**

What does the `switch` statement, with the `int` variable `n`, on the right print when

1) `n` is 0?

2) `n` is 2?

3) `n` is 4?

```
switch(n) {
    case 1:
        printf("solo\n") ;
        break ;
    case 2:
        printf("duet\n") ;
    case 4:
        printf("quartet\n") ;
        break ;
    default:
        printf("choir\n") ;
}
```