

NCSU ECE 209 Sections 602

Exam 1 Fall 2009

29 September, 2009

This is a closed book and closed notes exam. It is to be turned in by 5:45 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 (8 points)

Complete the following program so that it writes your name 100 times. Your name should be written on a separate line each of these 100 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.

ECE209	extra209	__X
Hello	209ECE	double
NCSU_ECE_109	EuroTo\$	#Chars

Problem 3 (3 points)

Under what conditions does the following C expression have the value of 1?

```
getchar() == EOF
```

Problem 4 (25 points)

Below is a sequence of C statements defining some variables

```
int    vii = 7 ;
char   chr = 'd' ; /* ASCII code for 'd' is 100 */
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.

chr + 2	
xhf * 2.0	
vii + 3	
vii * 3 / 5	
1 + 1.0	
vii / 10	
3 / vii * 5	
vii = 17	
(int) xhf	
0 1	
0 && 1	
vii % 5	
vii ? vii+1 : vii+2	
0 * xhf	
(double) vii	
vii++	
chr > xhf	
vii += 10	
vii - 0.0	
vii * 0.5	
xhf == (int) xhf	
0 <= 0 <= 0	
0 < 0 == 0	
vii != vii	
1 5 && 0	

Problem 5 (15 points)

Each of the following four `for` or `while` loops print numbers. For each loop (which are often preceded by a few initializations) 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.

Assume that `i` and `j` have already been declared as `int` variables before each loop.

Some of these are rather tricky.

```
for (i=1; i<5; ++i)
    printf("%d\n", i) ;
```


```
for (i=1, j=5; i<j; i=i+4, j=j+2)
    printf("%d %d\n", i, j) ;
```


```
i=6 ;
while(i) {
    i = i - 3 ;
    printf("%d\n", i) ;
}
```


```
i = 17 ;
while ((i=i+3)<25)
    printf("%d %d\n", i) ;
```


Problem 6 (10 points)

Suppose variables `j`, `k`, and `s` have been declared as follows:

```
int j;
char k;
char p[]; /* a character string */
```

In the table below there is a `scanf` call in each cell of the first column. The remaining columns are labeled `j`, `k`, and `p`. 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.

109 ece 209

For each row, in the cells of the columns labeled `j`, `k`, and `p`, 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>j</code>	<code>k</code>	<code>p</code>
<code>scanf("%d %s", &j, p)</code>	109	_____	"ece"
<code>scanf("%c", &k)</code>	_____	'1'	_____
<code>scanf("%1d%c%s", &j, &k, p)</code>			
<code>scanf("%d %c %s", &j, &k, p)</code>			
<code>scanf("%*d%2s%c", p, &k)</code>			
<code>scanf("%x %s", &j, p)</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 `n` asterisks. Thus if `n` contains 5, five asterisks will be printed. Use no variables other than `n` and `j`. [Problem taken from CodeLab homework.]

Problem 9 (5 points)

Write a little section of code that uses an `int` variable `n` (which has already declared) and prints "n is a positive even number" if `n` is positive and even. By the way, `n` is even when the remainder of dividing `n` by 2 is 0.

Problem 8 (10 points)

Suppose there is a double variable X with the value 209.602 and an int variable M with the value 109. 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.

Because blanks are so important in the target output, a shadowy happy face ☺ 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, 0x6D, 0155, and 'm' are other ways of saying 109 in C.

209.602☺☺109	printf("%g%5d\n", X, M) ;
☺☺209.60☺109	printf(" \n", X, M) ;
209.6020☺006d	printf(" \n", X, M) ;
2.096020e+02☺m	printf(" \n", X, M) ;
X=209.6, ☺M=109	printf(" \n", X, M) ;
☺☺210☺%☺155	printf(" \n", X, M) ;

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("single\n") ;
        break ;
    case 2:
        printf("double\n") ;
        break ;
    case 3:
        printf("triple\n") ;
        break ;
    case 4:
        printf("home run\n") ;
    default:
        printf("huh?\n") ;
}
```