EECS 110 Midterm
February 7, 2007

Don't panic! *Read each question through. If any part confuses you, ask me privately. Watch your time. Don't spend forever on any one question. Write cleanly. If you need to make big changes, X out your answer, write "see back" and write your new version on the back, with the number of the question.*

1. (5 pts) Show the output of the following program fragment:

Program	Output

`for (int i = 0; i <= 10; ++i) {`
`  printf("%d ", i);`
`  ++i;`
`}`
	`0 2 4 6 8 10`

Comment [CKR1]: The key part is noticing that i is incremented twice.

2. (5 pts) Show the output of the following program fragment:

Program	Output

`for (int i = 5; i > 0; --i) {`
`  for (int j = 0; j < 4; j += 2) {`
`  printf("%d %d\n", i, j);`
`  }`
`}`
	`5 0`
	`5 2`
	`4 0`
	`4 2`
	`3 0`
	`3 2`
	`2 0`
	`2 2`
	`1 0`
	`1 2`

Comment [CKR2]: Most common mistake: only outputting 2 pairs, usually, (5 0) (4 2)

3. (10 pts) `print_roi()` is supposed to read an investment amount and annual interest rate, and print the compounded value of the investment for 5 years, like this:

```
Enter investment and rate: 10000 7.2
Year    Value
----    -----
  1   10720.00
  2   11491.84
  3   12319.25
  4   13206.24
  5   14157.09
```

But it's full of mistakes (more than 6). Circle every error you can find. Write the correct code next to it. Write SYN if and only if the mistake will cause a compilation error. Write LOG if it's a logical mistake that will compile and run but crash or do the wrong thing.

```
print_roi()
{
    double invest, rate;
    printf("Enter investment and rate: ");
    scanf("%lf %lf", invest, rate);
    printf("Year    Value\n");
    printf("----    -----\n");
    for (i = 1, i < 5, +i)
        invest = invest + rate / 100;
        printf("%3d%11.2lf\n", i, invest);
}
```

Comment [CKR3]: SYN Add void return type.

Comment [CKR4]: LOG Add &

Comment [CKR5]: LOG Add &

Comment [CKR6]: SYN Add int type

Comment [CKR7]: SYN Replace with ;

Comment [CKR8]: LOG Replace with <=

Comment [CKR9]: SYN Replace with ;

Comment [CKR10]: LOG Replace ++

Comment [CKR11]: SYN if i is made a local FOR variable, LOG if i is made a local function variable. Add { and } around the body of loop

Comment [CKR12]: LOG Add "invest *"

Comment [CKR13]: BOTH Add }

Comment [CKR14]: Most common mistakes: the return type for print_roi(); making mistakes out of stylistic issues in declaring variables or printing

4. (5 pts) Write a `switch` statement that takes a character (upper or lower case) and prints "vowel" if it's a vowel (A, E, I, O, or U) and prints "consonant" otherwise.

```
switch(tolower(ch))
{
    case 'a': case 'e': case 'i': case 'o': case 'u':
        printf("vowel\n");
        break;
    default:
        printf("consonant\n");
}
```

Comment [CKR15]: toupper() also OK if capitals used in case's.

If neither, then cases must list both upper and lower case alternatives.

Comment [CKR16]: break is important

Comment [CKR17]: Most common bad move: separate prints for each vowel. Next most common: not handling upper and lower case.

5. (10 pts) A number has **odd parity** if it has an odd number of 1 bits in its binary representation. This is equal to how many odd numbers you get (including the number itself), if you repeatedly integer-divide by 2, until you hit 0.

Divisions	Odds	Parity
7, 7 / 2 => 3 , 3 / 2 => 1, 1 / 2 => 0	3	odd
17, 17 / 2 => 8, 8 / 2 => 4, 4 / 2 => 2, 2 / 2 => 1, 1 / 2 = > 0	2	even

Define the function `odd_parity()` to take a positive integer and return 1 if it's an odd parity integer and 0 if it's an even parity integer:

```
int odd_parity(int n) {
    int i = 0;
    while (n > 0) {
        i += n % 2;
        n /= 2;
    }
    return i % 2;
}
```

Comment [CKR18]: No need for another variable because n is local.

Comment [CKR19]: += is simpler than = and +.

Comment [CKR20]: No need for an IF.

Comment [CKR21]: /= is simpler than = and /.

Comment [CKR22]: No need for an IF.

Comment [CKR23]: Most common mistakes:

- reading in a value for n – n is a parameter so it has a value already
- printing the result – printing is not returning
- not counting n itself if odd
- not incrementing the loop variable in a FOR, e.g., for (int i = n; i > 0; i / 2) does NOT change i

I wrote "no IF needed" in many places, but I did not take points off for that.