ECE 131 Programming Fundamentals – Exam #1

Fall, 2011
Closed book, closed notes

Name:

8 pts.

1. What innovation did John von Neumann introduce in the EDVAC computer:

   (a) He replaced the vacuum tubes with transistors, and added a really sweet graphics card.

   (b) The idea of a stored-program, where the program is stored in computer memory, along with the data it operates on.

   (c) The first assembly language was introduced for the EDVAC instruction set.

   (d) A cache memory was added to the memory hierarchy.

10 pts.

2. List the following in decreasing order of access speed (i.e., fastest first): *cache memory*, *registers*, and *virtual memory*.

registers
Cache memory
virtual memory

10 pts. 3. Convert 64_{10} to a binary number.

64/2 = 32, rem 0
32/2 = 16, rem 0
16/2 = 8, rem 0
8/2 = 4, rem 0
4/2 = 2, rem 0
2/2 = 1, rem 0
1/2 = 0, rem 1
Answer: 1000000

10 pts. 4. Convert 10111_2 to a decimal number.

1 x 1 = 1
1 x 2 = 2
1 x 4 = 4
0 x 8 = 0
1 x 16 = 16

Answer: 23

10 pts. 5. Convert 10_{16} to a binary number.

By inspection:

Answer 10000

12 pts. 6. Given the following C variable declarations:

```
int a=1, b=2, c=3;
```

What do the following expressions evaluate to (a table of C operators is attached):

(a) `a + b * c` $1 + (2*3) = 7$

(b) `a * b + c` $(1*2) + 3 = 5$

(c) `(a + b) + c` $(1+2) + 3 = 6$

(d) `c % a` $3 \% 1 = 0$

(e) `a + b - a + b` $1+2-1+2 = 4$

(f) `a * b - a * b` $(1*2) - (1*2) = 2 - 2 = 0$

(g) `a + b * c % a` $1 + ((2*3) \% 1) = 1 + (6 \% 1) = 1 + 0 = 1$

(h) `(a != c) * c` $(1 != 3) * 3 = 1 * 3 = 3$

(i) `a += b *= c` $a += (b *= c) \Rightarrow 1 += (6) = 7$

(j) `a *= b += c` $1 *= (2 += 3) \Rightarrow 1 *= 5 = 5$

(k) `1/c * 3` $(1/3) * 3 = (0) * 3 = 0$

(l) `1.0/c * 3` $(1.0/3) * 3 = (0.333\ldots) * 3 \approx 1.0$

10 pts. 7. In the following C program, circle each error, and explain why it is an error:

```
main();
{
   char mychar;
   INT c=3; b=2;
   const int z=2;
   int 2z=4;
   mychar = z;
   z = b;
   mychar = z ** 2;
};
```

INT should be int ; should be ,

invalid identifier

z is const, not modifiable

** is not a C operator

10 pts. 8. What does the following program print? Explain your answer.

```
#include <stdio.h>
int x = 3;

main()
{
   int y = 2;
   printf("x = %d", x+y);
}
```

Output : x = 5

since x = 3 and y = 2, x+y = 5,
and that value is used in the printf output

10 pts. 9. What does the following program print? Explain your answer.

```
#include <stdio.h>
int x = 4;

main()
{
   int x = 7;
   printf("%d", 2 * x);
}
```

Output : 14
The x used in the printf statement is
the x defined in the main function block,
not the global x.

10 pts. 10. What does the following program print? Explain your answer.

```
#include <stdio.h>
int func();  // function prototype

main()
{
   int x;
   x = func() * func();
   printf("%d", x);
}

int func()
{
   static int x=0;
   x += 2;
   return x;
}
```

Output: 8

The x in func() is static, initialized to 0.
The first call to func() increments it to 2, then returns it.
The second call to func() increments it to 4, then returns it.
In main(), the line x = func() * func() then evaluates to x = 2 * 4
The value of x used in the printf statement is 8.

Operator	Description	Associativity
()	grouping	left to right
−	unary minus (negation)	right to left
+	unary plus	
++	increment	
−−	decrement	
sizeof	size of an object	
*	multiplication	left to right
/	division	
%	modulus	
+	addition	left to right
−	subtraction	
==	equality	left to right
! =	inequality	
=	assignment	right to left
* =	multiplication assignment	
/ =	division assignment	
+ =	addition assignment	
− =	subtraction assignment	