CIS 2107
Computer Systems and Low-Level Programming
Spring 2013
Midterm

March 21, 2012

Name: _____

Page	Points	Score
1	11	
2	13	
3	11	
5	9	
7	5	
7	27	
10	24	
Total:	100	

Instructions

The exam is closed book, closed notes. You may *not* use a calculator, cell phone, etc.

For each of the questions of this quiz, you can assume the following sizes for C data types:

type	**bytes**
char	1
short	2
int	4
long	8
float	4
double	8
void*	4

Some short answer questions

1. (1 point) In the unix shell, what's the command to change from the current directory to the parent of your current directory?

---

2. (1 point) In the unix shell, what's the command to create a directory called `stuff` and place it in your current directory?

---

3. (1 point) Now that you have the directory created in question 2), what's the command to take a file called `junk.txt`, which is in your current directory, and put it into the new directory?

---

4. (1 point) We've described the storage hierarchy in modern computers as a type of pyramid. What two things are true the farther up the pyramid we go?

---

5. (1 point) What's the program manipulates your program's text before it gets fed to the compiler?

5. _____

6. (1 point) What's the program that translates assembly language into machine language (*i.e.* a binary)?

6. _____

7. (1 point) What's the program that combines binaries to form an executable?

7. _____

exam continues...

8. Some conversions.
   (a) (1 point) 72 gbytes = ? kbytes

   (a) _____

   (b) (1 point) 2 hours = ? milliseconds

   (b) _____

   (c) (1 point) 112 gbits = ? tbits

   (c) _____

   (d) (1 point) 144 mbits = ? tbytes

   (d) _____

   (e) (1 point) 15 minutes = ? microseconds

   (e) _____

9. Convert 246_{10} to:
   (a) (2 points) base 2

   (b) (1 point) base 16

10. Using the approximation trick that we talked about in class, about how much are each of the following?
    (a) (1 point) 2^{41}

    (a) _____

    (b) (1 point) 2^{16}

    (b) _____

    (c) (1 point) 2^{29}

    (c) _____

11. (3 points) What is $11011110011_2 + 10011010_2$ in base 2?

	1	1	0	1	1	1	1	0	0	1	1_2
+				1	0	0	1	1	0	1	0_2

12. (3 points) What is $DEDC5B9_{16} + 61B3_{16}$ in base 16?

	D	E	D	C	5	B	9_{16}
+				6	1	B	3_{16}

13. **data representation**. For these questions, please remember to answer in hex, not binary.

    (a) (1 point) In hex, what is the smallest integer that can be represented by a 16-bit two's complement int?

    (a) _____

    (b) (1 point) In hex, what is the largest integer that can be represented by a 16-bit two's complement int?

    (b) _____

    (c) (1 point) In hex, what is the smallest integer that can be represented by a 16-bit unsigned int?

    (c) _____

    (d) (1 point) In hex, what is the largest integer that can be represented by a 16-bit unsigned int?

    (d) _____

    (e) (1 point) In hex, what is -1 as a 16-bit two's complement int?

    (e) _____

14. (6 points) **Some bit operations.** If we have `char x = 0x3D, y = 0xA7;`, what is the result of the following operations? Your answer must be in the form of exactly two hex digits[1].

(a) x|y

(a) _____

(b) x||y

(b) _____

(c) x<<2

(c) _____

(d) ~x

(d) _____

(e) ~~x

(e) _____

(f) x^y

(f) _____

(g) x &&1

(g) _____

[1]Ignore the possibility of promotion to 32-bit ints. Behave as though we're living in the land of 8-bit arithmetic.

(h) -x

(h) _____

(i) !!x

(i) _____

(j) x<1

(j) _____

(k) x&y

(k) _____

(l) x^y^y

(l) _____

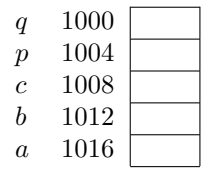
15. (4 points) What's printed by the following code?

```
3  int main(void) {
4    char c;
5    unsigned char uc;
6
7    c=uc=0x7F;
8
9    c+=1;
10   uc+=1;
11
12   printf("%d\n", c);
13   printf("%u\n", uc);
14
15   return 0;
16 }
```

16. (5 points) If I have the following:

```
1  int main(void)
2  {
3    int a=10;
4    int b=20;
5    int c=30;
6
7    int *p, *q;
8
9    p=q=&a;
10
11   b++;
12   q-=4;
13
```

and memory is laid out like this:



what do you see if you print:

(a) a

(a) _____

(b) &a

(b) _____

(c) b

(c) _____

(d) &b

(d) _____

(e) p

(e) _____

(f) *p

(f) _____

(g) &p

(g) _____

(h) q

(h) _____

(i) *q

(i) _____

(j) &q

(j) _____

points: _____
out of a possible 27

exam continues...

17. (5 points) How would 187.5625_{10} be stored in a C `float` variable?

18. (5 points) **Recognizing the value of a floating-point variable.** In this question, consider 6-bit floating-point numbers. What number is represented by the 0 10 000, where:

    - 0 is the sign bit
    - 10 is stored in the mantissa field
    - 000 is stored in the exponent field

19. (10 points) Write a function called `bit_is_set( )` which takes as arguments an `unsigned int x`, and an `int i`. The function checks to see if the ith bit of x is set and returns:

    - 1 if the ith bit is set to 1
    - 0 if the ith bit is set to 0
    - -1 if i is an invalid index.

    Indices start from the right: so right-most bit would be index 0. Do not assume anything about the size of integers.

20. (12 points) Write a function whose sole argument is a C string. The function removes all leading and trailing whitespace from the string. For example, if before your function is called, the string is "␣␣␣What a long exam␣␣␣␣␣", after the function is called, the string is "What a long exam". (Note: you're modifying the original string. You're not creating and returning a new string.) You may use any function in <`ctype.h`>, but do not use any functions in <`string.h`>.

21. (12 points) Write a function which is passed an `int start`, and an `int end`. The function returns an array of `int` consisting of all of the integers from `start` to `end` inclusive. If `start>=end` or on error the function returns a `NULL` pointer. It is up to the caller to free any memory allocated by your function. Do not use the `[ ]` operator in the body of your function. For example, if `start=5` and `end=11`, the function returns a pointer to {5,6,7,8,9,10,11}.

22. (12 points) Write a function which is passed the name of a file. The function prints everything read from the file, but capitalizes the first letter of every word. For example, if the text "this is a pretty LONG exam" is read, the program prints "This Is A Pretty LONG Exam". You may assume that the text includes only letters and spaces (as defined by the `isspace( )` function in <`ctype.h`>).