Sec: 1 2 (circle) ECE-1021 Fall 2004 Exam #3 (FINAL) 14 December 2004

NAME: KEY Score:______/100

Unless specified otherwise, assume that:

- The variables i, j, k, l, m, and n are declared as int
- The variables u, v, w, x, y, and z are declared as doubles.
- Any reference to a "floating point" value is to be taken as a type double.
- Any reference to an "integer" value is to be taken as a type int.

IEEE-574 Single-precision floating point standard: 32 bits total including an 8-bit exponent.
IEEE-574 Double-precision floating point standard: 64 bits total including an 11-bit exponent.

Multiple Choice (2 points each) – choose the best answer from among those offered.

1) How many bits are used by the standard ASCII code?

a) 4.
[✓ b)] 7.
c) 8.
d) 16.

2) All ASCII characters fall into exactly one of which of the following pairs of groups?

[✓ a)] Printing and Control.
b) Upper case and Lower case.
c) Alphanumeric and Punctuation.
d) Control and Graphical.

3) What is the maximum number of distinct values that can be represented with 24 bits?

a) 4,294,967,296
[✓ b)] 16,777,216 $= 2^{24}$
c) 65,536
d) 32,768

4) What must be true of a **switch()** statement?

[✓ a)] The controlling expression must return an integer result.
b) A default **case** is required and must appear last in the *case* list.
c) The only way to exit a **switch()** structure is through the use of a break statement.
d) Multiple values can be listed on a case line – such as **case 1,2,3**:

5) In the base-5 number 323.401, what is the weighting of the digit '4'?

$\downarrow$ 5^{-1}

a) 0.04
b) 1/20
[✓ c)] 5e-1
d) 5e-3

0:32

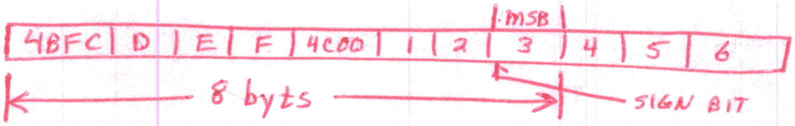
6) Which of the following is not one of the allowed ways to represent negative integers under the C Language Standard?

[✓ a)] Offset binary. (POSITIVE NUMBER NOT REPRESENTED USING PURE BINARY)
b) Signed binary.
c) One's complement.
d) Two's complement

7) An IEEE-double precision floating point value is stored at address 0x4BFC in Little Endian format. At what address can the byte containing the sign bit be located?

FROM EXAM FIRST PAGE: IEEE doble : 64bits = 8bytes

a) 0x4BF9.
b) 0x4BFC.
c) 0x4BFF.
[✓ d)] 0x4C03.



8) If it takes N digits to represent the integer value V in a positional number system using base B, which of the following relations will be true?

$1000 \leq xxxx < 10000$

$10^3 \leq xxxx < 10^4$

$B^{(N-1)} \leq$ N DIGITS $< B^N$

a) V = B^N
b) (B^N < V) AND (V <= B^(N+1))
[✓ c)] (B^(N-1) <= V) AND (V < B^N)
d) V^N = B

9) Which of the following is not a requirement of the *case* labels in a *switch()* construct?

a) They must unique. T
[✓ b)] They must evaluate to valid ASCII codes. F
c) They must be integers expressions. T
d) They must be constant expressions. T

10) Given a bit pattern representing a floating point value in the standard IEEE format, how can the negative of that value be represented?

a) By taking the two's complement of the mantissa.
b) By inverting the sign bit and all bits of the mantissa.
c) By treating the entire pattern as though it were an integer and taking the two's complement of it.
[✓ d)] By inverting the sign bit.

4:00

Questions 11-15 refer to an IEEE-754 Single Precision floating point representation.

11) What representation is used for the exponent?

a) signed binary.
[✓ b)] offset binary.
c) one's complement.
d) two's complement.

12) How many bits of the mantissa are stored?

a) 16.
[✓ b)] 23.
c) 24.
d) 33.

$\underset{\text{WIDTH}}{32} - \underset{\text{SIGN BIT}}{1} - \underset{\text{EXPONENT}}{8} = \underset{\text{MANTISSA}}{23}$

13) In what order are the components that make up the value stored?

[✓ a)] (sign)(exponent)(mantissa)
b) (sign)(mantissa)(exponent)
c) (exponent)(sign)(mantissa)
d) (mantissa)(exponent)(sign)

14) If the representation for positive infinity were stored in Little Endian format, what would be stored in the four bytes, beginning with the base address?

[✓ a)] [00][00][80][7F]
b) [00][00][00][FF]
c) [7F][80][00][00]
d) [FF][FF][FF][FF]

$+\text{inf} : \underset{S}{0}\ \underbrace{11111111}_{\text{exp}}\ \underbrace{000 \longrightarrow 0}_{\text{MANT}}$

$\underset{\text{MSB}}{7F}\ 80\ 00\ \underset{\text{LSB}}{00} \xrightarrow[\text{ENDIAN}]{\text{LITTLE}} 00\ 00\ 80\ 7F$

15) A floating point value is represented by the pattern 0x3EC00000. What is the value in decimal?

a) -0.75×2^{-16}.
b) -0.375.
[✓ c)] $+0.375$.
d) $+0.75 \times 2^{16}$.

3 E C 0 0

$\underset{\substack{S \\ \uparrow \\ \text{POSITIVE}}}{0}\ |\ \underset{\text{exp}}{011\ 1110\ 1}\ |\ \underset{\text{MANT}}{100\ 0000 \ldots\ldots 0}$

exp: 7D

$7 \times 16 = 112$, $112 + 13 = 125$

$125-127=-2$

IMPLIED 1

$+\underbrace{1.10000_2}\times \underbrace{2^{-2}}$

$+\left(1+\frac{1}{2}\right)\times\left(\frac{1}{4}\right)$

$+\frac{1.5}{4} = \underline{\underline{0.375}}$

7:44

16) In an IEEE-754 representation, why does an exponent consisting of all zeros and an exponent consisting of all zeros except for a trailing one represent the same exponent value?

[✓ a)] To avoid a gap as the representation moves from a normalized to a non-normalized interpretation.
b) To increase the static range of the representation.
c) So that zero may be exactly represented.
d) It doesn't.

17) The combination of a structure and a set of functions that act on it are a precursor to what?

a) A module.
[✓ b)] The concept of an "object" in C++.
c) A function structure.
d) A compound structure.

18) What is not an advantage of defining a small set of "primitive" functions that are the only functions allowed to directly operate on a structure's elements and then requiring that all other functions that need access do it only via these primitives?

a) The details of how data is organized within the structure can be abstracted (a.k.a., hidden) from the User. ADV
b) Changes to how the data is organized within the structure will only require modifications to the primitive functions. ADV
[✓ c)] A small set of primitives results in more efficient code in terms of execution speed. FALSE
d) The frequently confusing task of structure dereferencing is limited to a small number of functions, namely the primitive functions. ADV

19) What is the practical difference between a **`typedef`** statement and a similar **`#define`** statement?

a) A **`#define`** statement cannot be expanded on a variable declaration line. F
[✓ b)] A **`#define`** may not behave as expected if a pointer is part of the definition. T
c) There is no comparison – they are completely different things. F
d) There is no difference. F

20) Which of the following pairs of statements are identical?

[✓ a)] **`(*ptr).element`** AND **`ptr->element`**
b) **`(*ptr)->element`** AND **`ptr.element`**
c) **`*ptr.element`** and **`ptr->element`**
d) **`*ptr->element`** AND **`ptr.element`**

9:42

21) Which of the following is not a difference between structures and arrays?

a) Structures are passed by value while arrays are passed by reference. DIFF
b) Structures are not limited to containing variables of a single data type. DIFF
[✓ c)] Structures can contain elements that are pointers. EITHER CAN
d) Structures and arrays use different operators to access their elements. DIFF

22) Which of the following types of I/O operations are normally carried out on text files?

a) character, formatted, and unformatted.
b) string, formatted, and unformatted.
c) character, string, and unformatted.
[✓ d)] character, string, and formatted.

23) The functions **fset()**, **fread()**, and **fwrite()** are typically used with files opened for

a) text operations.
[✓ b)] binary operations.
c) append operations.
d) random access operations.

24) The value stored in a pointer variable represents

a) The initial memory location of the program's data segment. F
[✓ b)] The address at which some piece of information is stored. T
c) The address of the beginning of a text string. NOT ALWAYS
d) The address of the beginning of an array. NOT ALWAYS

25) What is the difference between the following two statements?

```
#include <file.h>
#include "file.h>
```

a) The <> version can only be used for standard library files. F
[✓ b)] The "" version looks for the file in the specified path or current directory first. T
c) The "" version is for C++ header files and the <> version is for C header files. F
d) There is no difference, both are allowed and are equivalent. F

26) In a function call, the actual parameters are separated by

[✓ a)] commas.   eg: z = pow(x, y);
↑
b) semicolons.
c) colons.
d) spaces.

27) What is the proper format specifier for an argument of type **`double`**?

a) **`%d`** in both **`scanf()`** and **`printf()`**.
[✓ b)] **`%f`** in **`printf()`** and **`%lf`** in **`scanf()`**.
c) **`%lf`** in both **`printf()`** and **`scanf()`**.
d) **`%f`** in **`printf()`** and **`%d`** in **`scanf()`**.

28) What are the three basic building-block structures of a structured program?

a) Input statements, output statements, and computation statements.
b) Goto's, loops, and I/O.
[✓ c)] Sequences, selections, and repetitions.
d) Looping structures, switching structures, and computed goto statements.

29) Why does C offer three different looping structures?

a) Because no one looping structure can implement all of the different types of loop logic that might be needed by a program. F
b) Because the **`for()`** loop is only capable of executing a finite and predetermined number of times and the **`while()`** loop is not guaranteed to execute the loop code at least once. F
[✓ c)] Because each structure lends itself to a certain type of looping logic and program readability and maintainability are enhanced if the structure used matched the logic implemented. T
d) For compatibility with other languages. F

30) Modulo division of m by n (i.e., **`m%n`**) is used to return

a) The result of integer division. F
b) A value of 1 if m can be evenly divided by n. F (WILL BE ZERO)
[✓ c)] The same value that would be obtained by **`m - n*(m/n)`**.
d) The result of a floating point division even if the operands are integers. F

Eg m % n
5 % 3 is 2

m - n*(m/n)
$5 \quad 3*\underbrace{(5/3)}$

$\underbrace{5 - 3*1}_{2}$ ← INT DIV.

15:38

True/False (4 pts each)
NOTE: You ***must*** explain your reasoning to receive full credit. T/F along will only receive 1pt.

31) (T / [✓ F]) If a file open operation using `fopen()` fails, then the file should be immediately closed using `fclose()`.

NOTHING ELSE SHOULD BE DONE WITH THAT FILE SINCE NO FILE WAS OPENED.

32) (T / [✓ F]) The functions `fread()` and `fwrite()` need to know what type of data is being referenced.

THEY PERFORM DIRECT TRANSFERS BETWEEM MEMORY AND A FILE. NO FORMATTING, TRANSLATION, OR MANIPULATION IS PERFORMED.

33) (T / [✓ F]) A recursive function need not have a non-recursive path through it in order to be successful.

IF IT LACKS A NON-RECURSIVE PATH, AN INFINITE LOOP WILL RESULT.

34) (T / [✓ F]) When using `printf()`, printing out a percent sign in the format string requires that it be escaped with a backslash just like a newline character or a double quote.

MUST USE THE FORMAT SPECIFIER %%.

THE ESCAPE SEQUENCE IS TO PERMIT THE USE OF CHARACTERS THAT THE COMPILER MIGHT MIS-INTERPRET.

35) ([✓ T] / F) When using `printf()`, the same format specifier is used for both a float and a double.

THE COMPILER AUTOMATICALLY CONVERTS ALL ARGUMENTS to printf() THAT ARE OF TYPE float TO TYPE double, HENCE A SINGLE FORMAT SPECIFIER IS ADEQUATE.

20:00

Longer Answer (5 pts each)

36) Fill in the following table with the decimal value represented by each bit pattern using the representation at the top of the column.

PATTERN	Pure Binary	Two's Compl	One's Compl	Signed Binary	Offset Binary
000	0	0	0	0	-4
001	1	1	1	1	-3
010	2	2	2	2	-2
011	3	3	3	3	-1
100	4	-4	-3	-0	0
101	5	-3	-2	-1	1
110	6	-2	-1	-2	2
111	7	-1	-0	-3	3

21:37

37) Write a recursive function that computes S(n) = 2 + 4 + 6 + ... + 2n [Hint: S(n) = S(n-1) + 2n]

```
int s(int n)
{
    if (n<1)
        return n;
    return S(n-1) + 2*n;
}
```

if n=2, 2n=4

∴ S(2) = 2+4

HENCE S(1) = 2 (ALT BASE CASE)

S(0) = 0 BASE CASE

24:14

38) A particular number is represented using two bytes in signed binary, offset binary, and two's compliment format. In no particular order, the three representations yield values of **0x31E0**, **0xB1E0**, and **0xCE20**. What is the original base-10 number?

	SIGN BIT		
SIGN	SIGNED BINARY	OFFSET BINARY	TWO'S COMPL
+	0	1	0
−	1	0	1

PATTERN	SIGN BIT
31E0	0
B1E0	1
CE20	1

VALUE IS NEGATIVE

31E0 IS OFFSET

IN OFFSET BINARY

(ACTUAL VALUE) = (PURE) − (OFFSET)

PURE: 31E0

$3 \times 16 = 48$, $48 + 1 = 49$

$49 \times 16 = 784$, $784 + 14 = 798$

$798 \times 16 = 12768$, $12768 + 0 = 12768$

OFFSET: 0x8000 → 2^{15} → 32768

ACTUAL VALUE = 12768 − 32768

= −20,000

28:14

39) Write a complete C function called **log_b()** that takes two floating point arguments. The function returns the logarithm of the first number using the second number as the base. The base is assumed to be strictly positive.

```
#include <math.h> /* log() */

double log_b(double x, double b)
{
    return log(x) / log(b);
}
```

$1000 = 10^3$

$\therefore$ 3 IS BASE 10 log OF 1000

IF y IS BASE b log OF x

THEN $x = b^y$

$\ln(x) = y \cdot \ln(b)$

$y = \ln(x)/\ln(b)$

29:58

EXTRA CREDIT #1: (5 pts)
A flag pole is used to transmit a value by hoisting six colored flags. Each flag is one of the following colors – red, orange, yellow, green, blue. The value of each color is as follows: R = 0; O = 1, Y = 2; G = 3, B = 4. The value is read from top (most significant) to bottom and correspond to left (most significant) to right in the tables below.

a) How many different values can be represented by this signaling system? 15,625

$5^6 = 15625$

b) If the flags represent an unsigned integer, what decimal value is represented by the following configuration of flags? 5555

	FLAG 5	FLAG 4	FLAG 3	FLAG 2	FLAG 1	FLAG 0
	ORANGE	GREEN	BLUE	YELLOW	ORANGE	RED
DIGIT:	1	3	4	2	1	0

1 ⇒ $1\times5+3 = 8$ ⇒ $8\times5+4=44$ ⇒ $44\times5+2=222$ ⇒ $222\times5+1=1111$ ⇒ $1111\times5+1$ = 5555

c) Circle the color of each flag if the decimal value of the number being displayed is 2500.

FLAG 5	FLAG 4	FLAG 3	FLAG 2	FLAG 1	FLAG 0
B G Y O [✓ R]	[✓ B] G Y O R	B G Y O [✓ R]	B G Y O [✓ R]	B G Y O [✓ R]	B G Y O [✓ R]

5 |2500
5 |500 r 0
5 |100 r 0
5 |20 r 0
5 |4 r 0
0 r 4

40000_5

d) Using the same reasoning that led to 2's compliment for representing signed values in binary, what color would each flag be if –2500 is displayed using 5's compliment?

FLAG 5	FLAG 4	FLAG 3	FLAG 2	FLAG 1	FLAG 0
[✓ B] G Y O R	B G Y [✓ O] R	B G Y O [✓ R]	B G Y O [✓ R]	B G Y O [✓ R]	B G Y O [✓ R]

L

BASIC CONCEPT ! $(-A)+A = B^N = 5^6$

$(-A) = 5^6 - A = 15,625 - 2500 = 13125$

5 |13125
5 |2625 r 0
5 |525 r 0
5 |105 r 0
5 |21 r 0
5 |4 r 1
0 r 4

410000_5

e) Using the same reasoning that led to offset binary for representing signed values in binary, what color would each flag be for the value zero?

FLAG 5	FLAG 4	FLAG 3	FLAG 2	FLAG 1	FLAG 0
B G [✓ Y] O R	B G [✓ Y] O R	B G [✓ Y] O R	B G [✓ Y] O R	B G [✓ Y] O R	B G [✓ Y] O R

BASIC CONCEPT

PURE	VALUE
0	-7812
7812	0

15624 + 7812

VALUE = PURE - OFFSET

$\frac{15624}{2} = 7812$

5 |7812
5 |1562 r 2
5 |312 r 2
5 |62 r 2
5 |12 r 2
5 |2 r 2
0 r 2

222222_5

42:30

EXTRA CREDIT #2 (5 pts)

The C Language Standard does not require that a compiler use the ASCII character set. Instead, it sets down minimum requirements for whatever character set is used. For instance, the character set must include all of the decimal digits and all of the uppercase and lowercase characters. However, not all of the punctuation characters that are included in the ASCII character set have to be supported by a given compiler in order to be ANSI-C compliant. For each of the 32 punctuation characters in the ASCII character set, indicate whether it is required or not by the C Language Standard. If it is required indicate at least one use that it is needed for.

Character	Use	Character	Use

!	LOGICAL NOT	;	END OF STATEMENT
			for( )

"	STRING LITERAL #include	<	LESS THAN, LTE #include
#	PREPROCESSOR DIRECTIVES	=	ASSIGNMENT RELATIONAL
$	NOT REQUIRED	>	GREATER THAN, GTE #INCLUDE STRUCT INDIRECTION
%	MODULO FORMAT STRINGS	?	CONDITIONAL
&	ADDRESS OPERATOR BITWISE AND LOGICAL AND	@	NOT REQUIRED
'	CHARACTER CONSTANTS	[	ARRAY INDEX START
(	FUNCTION EXPRESSION START ORDER OF OPS OVERRIDE TYPE CASTS	\	ESCAPE SEQUENCE LINE CONTINUATION
)	FUNCTION EXPRESSION END ORDER OF OPS OVERRIDE TYPE CASTS	]	ARRAY INDEX END
*	MULTIPLICATION DEREFERENCE BLOCK COMMENT DELIMITER	^	BITWISE-XOR
+	ADDITION POSITIVE SIGN INCREMENT OPERATOR	_	ALLOWED IN IDENTIFIER NAMES
,	FUNCTION ARGUMENT DELIMITER SUB EXPRESSIONS	`	NOT REQUIRED
-	SUBTRACTION NEGATIVE SIGN DECREMENT OPERATOR STRUCT INDIRECTION	{	COMPOUND STATEMENT START
.	DECIMAL POINT STRUCT MEMBER	|	BITWISE-OR LOGICAL-OR
/	DIVISION BLOCK COMMENT DELIMITER	}	COMPOUND STATEMEN END
:	CONDITIONAL CASE LABEL	~	BITWISE-NOT

52:00