

2021 Fall CPSC 240-1 Answers

Midterm #2 Concepts Test

November 9, 2021 2:30pm-4:30pm

Read me

Place your answers in the space following each question.

Place your name in the test in this front page and on the last page.

Near 4:25pm begin to save your test document in either odt, or doc, or docx format.

Before 4:29pm send your document as an attachment to me: holliday@fullerton.edu

If you encounter a question where you feel that you must guess an answer then place the word "Blank" in the space for the answer and you will receive 20% of the credit for that question.

If the answer space is empty then the points for that question are zero.

You may use any word processing tool at your disposal provided it can save files in one of the three accepted formats.

If your computer has no word processor program then try Google Docs, which saves files in every format ever created on this planet.

The total point value of this test is 100 points, which is one-sixth of your course grade.

Every effort has been made to create unambiguous questions. If a question is truly ambiguous send me ordinary email or a chat message to ask for clarification. I will be at computer during the test period probably answering the backlog of email.

This is an open note test.

If your word processor does not display the colors correctly, then view the document in Google docs.

Proceed to the next page.

1. Convert $97\frac{5}{7}$ to an IEEE 64-bit hex number. [11]

Show sufficient intermediate steps so that the grader will be convinced the answer was not simply copied from a calculator.

Begin solution:

$97 = 1100001$; //That much is easy

$$(5/7) \times 2 = 1 + (3/7)$$

$$(3/7) \times 2 = 0 + (6/7)$$

$$(6/7) \times 2 = 1 + (5/7)$$

$$(5/7) \times 2 = 1 + (3/7)$$

The sequence above is starting to repeat. There is no need to continue further

$$97\frac{5}{7} = 1100001 . 101101101101101101101101 \dots \times 2^0$$

$$= 1.100001101101101101101101101101 \times 2^6$$

$$= 1.1000 \ 0110 \ 1101 \ 1011 \ 0110 \ 1101 \ 1011 \ 0110 \ 1101 \ 1011 \dots \times 2^6$$

Add bias number and true exponent to get $\text{EFF} + 6 = 405 = \text{stored exponent}$.

Now we are ready to write our number in IEEE form:

0x4058 6DB6 DB6D B6DB

2. Convert this IEEE number to an equivalent number in base 10 in simplest form. [11]
0XC03F C800 0000 0000

Show sufficient intermediate steps so that the grader will be convinced the answer was not simply copied from a calculator.

Begin solution:

Be cautious: The first digit of the three digits C03 is C = 1100. The 1 on the left side indicates that this number is negative. We take away the left 1 and we see the stored exponent is 403.

Subtract: stored exponent – bias number = $403 - 3FF = 4 = \text{true exponent}$.

Significand is FC80 ...0 = 0.1111110010000000 ...0

Include the hidden bit: 1.1111110010000000 ...0

The positive version of our number is $1.1111110010000000 \dots 0 \times 2^4$

= $1.1111110010000000 \dots 0 \times 2^4$

= $1111111001.0000000 \dots 0 \times 2^{-5}$ The dot moved 9 places to the right, and then the exponent decreases by 9

= 1111111001×2^{-5}

= 1017×2^{-1} <== Calculator was used.

= $1017/2$

= 31.78125 <== Calculator was used

But we are not done. The original number was negative, therefore, the final answer is

-31.78125 <== It is negative

3. This is a twos complement integer: 0xB7A9 31FC 467E 8D20. What is its negative expressed in hex? [4]

Show sufficient intermediate steps so that the grader will be convinced the answer was not simply copied from a calculator.

Answer: Write the starting number below. Directly below each digit write its complement.

0xB7A9 31FC 467E 8D20

0x4856 CE03 B981 72DF

Now add 1

+1

0x4856 CE03 B981 72E0

4. What famous computer scientist founded the movement toward free open source software? [3]

Answer: Richard Stallman

Reference: https://en.wikipedia.org/wiki/Richard_Stallman

5. Explain to another assembly programmer how cwde works. [6]

Answer: The instruction cwde applies only to the register rax.

The lowest 16-bits of rax are a word of data. Using the rules of twos complement integers those 16 bits can be interpreted as a 16-bit signed integer.

The lowest 32 bits of rax are a double word of data.

The instruction cwde finds suitable values for bits numbers 31 -16 so that the entire double word is mathematically equal to the original word.

Lets do it by example.

Here is rax = 

The green area is the low word. There are 16 bits in the green area. The rules of twos complement will tell us the decimal value of those 16 bits. Let's say the value is 741.

Consider the 16-bit region immediately higher that the green word: the orange rectangle below.

rax = 

The action of cwde is to find new bits to place into the orange area so the numeric value of the entire low 32 bits shown in orange below has numeric value 741:

rax = 

6. What is stored in the frontend quadword of an inactive activation record? [6]

Answer: The value from rip at the instant the call was made to another function.

7. Suppose a C++ function is going to execute the “return;” statement. What internal changes will happen due to executing this one statement? [6]

Answer: These actions in succession:

pop rbx
pop rip

8. A function A is going to call another function B, where B is written in X86. What happens internally due to that function call. [6]

Answer: These actions in succession

push rip
move into rip the starting address of the called function

Further explanation:

“push rip” closes the AR of the function making the call to a new function

Moving a starting address into rip is outside the control of even the assembly programmer. This action is in the executable file itself. When the linker combines object files a smooth transition from executing statements of the caller function to executing statements of the called function.

9. What is inline assembly? State one valid reason why a software developer may want to use inline assembly. [Question 9 needs two answers.] [6]

Answer 1: Inline assembly is a technique of placing a block of assembly statements directly in the source code of a C++ or C function.

Answer 2:

First reason: The reason may be there is no assembler software (like Nasm) available on the developer's work station.

Second reason: The reason may be that the block of inline assembly is very small compared to the total size of the project, therefore, the programmers believe that putting the block of assembly in a separate file and linking in that separate file into the program is simply not worth the effort.

10. I have a C++ function stored in a file named `cauchy.cpp`. Show clearly how to translate that C++ source file into an equivalent assembly source file. [6]

Answer: `g++ -m64 -std=c++17 -c -S -o cauchy.o cauchy.cpp`

11. What is the gdb command that will output the contents of the low half of `xmm9` in floating point decimal format? [7]

Answer: `p/f $xmm9.v2_double[0]`

Wrong answer `x/1fg $xmm0.v2_double[0]`

12. A C++ function declares the following array: `float machine[20];`
What is the gdb command that will display the contents of cell number 6? [7]

Answer: Since the array contains only floats it is clear that a number in floating point format should be displayed.

`p/f machine[6]` `<== Wow this stuff is easy.`

Alternate answer: `x/1fg &machine[6]`

13. An assembly function makes the following declaration: [7]

```
segment .bss
bottles resq 30
```

What is the gdb command that will show the address where the 30 quadwords begin.

Answer: `p/x &bottles`

14. [Use the same array from the previous question.] What is the gdb command that will show the numbers in the first 4 quadwords of the array in floating point decimal format. [7]

Answer: `x/4fg (double *)&bottles`

15. The Cstring was declared in a C++ function: `char myname[30];`
The null-terminated character string has already been read into myname. What is the gdb command that will output that string. Avoid outputting ascii values. Here we only want printable chars. [7]

Answer `p/s myname`

That's it: 15 questions.

Comment about GDB questions

All of the GDB answers were checked in a live running program. Sometimes a GDB question will have two correct answers. Check your answers by testing each one in an executing program. If you have a second answer that outputs the same data as the answer shown in this document then let me know about it. You may even gain back some lots points.

Put your name and email address (CWID is optional) on this test in two places.

If the time is not yet 4:30pm then go back and review your answers. You have the entire internet to look for answers.

On or before 4:30pm send me your test document in one of three formats: odt, doc, or docx.

Only 3 formats are acceptable. PDF is NOT one of them. PDF documents will not be read.

`holliday@fullerton.edu`