

2025 Spring CPSC 240-3

Concept Test March 12, 2025: 10:00am-12:00noon

Basic info

The is a 2-hour test. Place your answer for each question in the space directly below the question. You are free to use a word processor to enlarge the space for your answer or reduce the size of the space as given.

The final document must be submitted in one of the three formats: doc, docx, or odt. All other formats will not be read and a zero score will be recorded.

Record your name and other information in the conclusion at the end of this test.

If X86 is required in the answer of question, then each X86 instruction used must appear in the published list of approved instructions.

If you cannot find an answer for any question, then write the word "Blank" without quotes in the space for the answer and you will receive 20% of the credit for that question.

Tests received with a time stamp after 12:05am will not be read.

The terms IEEE and IEEE754 are used interchangeably.

If a questions requires an algorithmic process to obtain the answer, then give sufficient intermediate steps so that the grader becomes convinced that you know the algorithm.

This is an open notes test. You may use any resource to obtain a answer except obtain a classmate's answer. That last action will have dire consequences if discovered.

You may find space insufficient for your answer. You may use your word processor to increase the amount of space for an answer if needed.

Continue to next page

1. Block answer. Let r9 be the number of Pepsi bottles on an advancing conveyor belt which packs the bottles into cartons for shipment to retail stores. Each carton holds r13 number of bottles. Make a block of clear concise assembly that will compute the number of filled cases that will be produced (r14) and the number of left-over bottles (r15) insufficient to fill a full case.

; r9 contains the total number of bottles, r13 the carton capacity.

```
mov  rax, r9    ; move total into rax
xor   rdx, rdx   ; clear rdx
div   r13        ; divide rax by r13 (quotient in rax remainder in rdx)
mov   r14, rax   ; r14 gets the number of full cases
mov   r15, rdx   ; r15 gets the number of leftovers
```

2. Who leads a chip manufacturing company as both CEO and chief of technology?

Lisa Su

3. Who first promoted the use of open source software and continues to promote non-proprietary software to this day?

Richard Stallman

4. Make a macro named stored_exp. The macro has an incoming parameter which is an xmm register. It has an outgoing parameter which is the stored exponent returned in a general register. An example of invoking the macro is this:

```
stored_exp xmm3 r10
```

```
stored_exp macro xmm_reg, reg
```

; Move the 64-bit double from xmm_reg to a general-purpose register

```
movsd  xmm0, xmm_reg    ; move double to xmm0
cvtsd2si reg,  xmm0     ; convert double to intr in reg
```

; for a normalized double the exponent occupies bits 52–62

```
shr     reg, 52          ; shift exponent to lower bits
and     reg, 0x7FFF      ; mask out only the 11 exponent bits
endm
```

5. There is an integer number in rdi. Show assembly instruction(s) that will flip bit number 9 to its complement.

```
xor    rdi, 0x200
```

(16 digits after adding a 0)

6. Convert this IEEE float number to a decimal float. 0X000CC00000000000

Break the number into parts

64-bit IEEE double = 1 sign bit, 11 exponent bits, 52 fraction bits.

0x000CC00000000000 in binary starts with 0 (positive).

Find the exponent

The stored exponent bits come out as 1.

True exponent = $1 - 1023 = -1022$.

Find the fraction

the fraction bits start with “100...” (from the C in 0x000C

the val is $1 + (F/2^{52})$ where F comes from the nonzero bits

$$= 1 + (1/2) + (1/16) + (1/32) = 1.53125$$

Answer

work form: $(1 + F/2^{52}) \times 2^{(-1022)}$

decimal: $2.225 \times 10^{(-308)}$.

7. What is a gdb command that may have produced the following output. There are several correct answers. You need one of them.

```
0x7fffffffdf20: 0x0000 0x0000 0x0000 0x0000 0xe083 0xf7bf 0x7fff 0x0000
0x7fffffffdf30: 0x6b80 0xf7dc 0x7fff 0x0000 0xe018 0xffff 0x7fff 0x0000
0x7fffffffdf40: 0x1c00 0x0001 0x0001 0x0000 0x11e6 0x0040 0x0000 0x0000
0x7fffffffdf50: 0x1880 0x0040 0x0000 0x0000 0xe2ed 0x1f57 0xf007 0x4daa
0x7fffffffdf60: 0x1100 0x0040 0x0000 0x0000 0xe010 0xffff 0x7fff 0x0000
0x7fffffffdf70: 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
0x7fffffffdf80: 0xe2ed 0xa137
```

When you have a good answer run a program in gdb mode. Enter your answer for this question. Make a screen shot of your program execution showing your input command to gdb and showing gdb's output. Paste that screen shot in the space below.

x/32h 0x7fffffffdf20

Screenshot below

```

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./question7...
(gdb) (gdb) run
Undefined command: "". Try "help".
(gdb) clear
No source file specified.
(gdb) run
Starting program: /mnt/c/Users/Riley/Desktop/cpsc-240-midterm/question7
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Memory allocated and initialized.
[Inferior 1 (process 1240) exited normally]
(gdb) x/32h 0x7fffffffdf20
0x7fffffffdf20: Cannot access memory at address 0x7fffffffdf20
(gdb) break main
Breakpoint 1 at 0x55555555178: file question7.c, line 3.
(gdb) run
Starting program: /mnt/c/Users/Riley/Desktop/cpsc-240-midterm/question7
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at question7.c:3
3   int main() {
(gdb) x/32h 0x7fffffffdf20
0x7fffffffdf20: 29541  29803  28783  25391  29552  11619  13362  11568
0x7fffffffdf30: 26989  29796  29285  12141  30065  29541  26996  28271
0x7fffffffdf40: 55      20296  21587  22868  17744  30781  13880  13919
0x7fffffffdf50: 52      16716  18254  17213  21806  18004  14381  20480
(gdb) continue
Continuing.
Memory allocated and initialized.
[Inferior 1 (process 1243) exited normally]
(gdb) █

```

8. What is the gdb command(s) that will show the quadwords between the address in rsp and the address in rbp? Hint: this answer may require two gdb commands.

When you have a good answer run a program in gdb mode. Enter your answer as a command. Verify that your answer produces an error-free positive integer. Make a screen shot of the input to gdb and the response from gdb. Paste the screen shot in the space below.

x/ (rbp - rsp) / 8gx \$rsp

Using host libthread_db library `"/lib/x86_64-linux-gnu/libthread_db.so.1"`.

Breakpoint 1, `function1 ()` at `question8.c:4`

4 `int local_var1 = 10;`

(gdb) info registers

rax	0x0	0
rbx	0x7fffffffdb68	140737488346216
rcx	0x555555557db8	93824992247224
rdx	0x7fffffffdb78	140737488346232
rsi	0x7fffffffdb68	140737488346216
rdi	0x1	1
rbp	0x7fffffffdb30	0x7fffffffdb30
rsp	0x7fffffffdb20	0x7fffffffdb20
r8	0x0	0
r9	0x7ffff7fca380	140737353917312
r10	0x7fffffffdb60	140737488345184
r11	0x203	515
r12	0x1	1
r13	0x0	0
r14	0x555555557db8	93824992247224
r15	0x7ffff7ffd000	140737354125312
rip	0x555555555175	0x555555555175 <function1+12>
eflags	0x202	[IF]
cs	0x33	51
ss	0x2b	43
ds	0x0	0
es	0x0	0
fs	0x0	0
gs	0x0	0
fs_base	0x7ffff7da0740	140737351649088
gs_base	0x0	0

(gdb) (gdb) p (\$rbp - \$rsp) / 8

Undefined command: `""`. Try `"help"`.

(gdb) p (\$rbp - \$rsp) / 8

`$1` = 2

(gdb) x/4gx \$rsp

`0x7fffffffdb20`: 0x0000000000000000 0x00007ffff7fe5af0

`0x7fffffffdb30`: 0x00007fffffffdb40 0x00005555555551da

(gdb) quit

A debugging session is active.

Inferior 1 [process 1753] will be killed.

Quit anyway? (y or n) ☐

9. Convert 19.8 to IEEE 64-bit hex number

As usual show sufficient work to convince the grader that you know IEEE numbers.

Convert 19.8 to IEEE 754 64-bit Hex

Convert to Binary

19 in binary is 10011

0.8 in binary is 0.1100110011001100110011001100110011001100110011001100110011001101 (rounded to 52 bits)

So, 19.8 in binary is 10011.1100110011001100110011001100110011001100110011001100110011001101

Step 2: Normalize

Shift decimal to get $1.0011110011001100110011001100110011001100110011001100110011001101 \times 2^4$

Step 3: Compute Exponent

Bias = 1023

True exponent = 4

Stored exponent = $4 + 1023 = 1027$

1027 in binary is 10000000011

1027 in hex is 0x403

Step 4: Compute Mantissa

Remove leading 1 from normalized value

Take next 52 bits: 0011110011001100110011001100110011001100110011001100110011001101

Convert to hex: 333333333333

Final Answer

IEEE 754 64-bit hex: 0x4033333333333333

10. A client in a legal office is putting information into a form. The instruction says "Enter your age. Enter control+d for decline to state. Age information is stored in the 64-bit variable years declared in the .bss segment: age resq 1

If the client enters an integer, store that integer in age. If the client enters control+d then store -1 in age.

Create a block of assembly that will perform the process of inputting an age number according to the requirements stated above. We assume that all inputs are valid integers.

```
; Assume: age is declared in the .bss (age resq 1)
; and we have a buffer defined for input.
mov    rax, 0      ; syscall number for read
```

```

mov    rdi, 0      ; file descriptor 0 (stdin)
mov    rsi, buffer ; address of input buffer
mov    rdx, 32     ; read up to 32 bytes
syscall                ; perform read; rax = number of bytes read
cmp    rax, 0
je     store_minus_one ; if no bytes were read (EOF, control+d)
; Otherwise, convert the input string to an integer.
; (Assume an atoi routine is available; result in rax)
call   atoi
mov    [age], rax   ; store the converted integer in age
jmp    done
store_minus_one:
mov    qword [age], -1 ; store -1 if EOF was encountered
done:

```

11. Consider this array declaration: `light dq 13, -49, 22, -41, 7, 12, 89, -13, 36`

What is a GDB command that will output the first 5 numbers of light in decimal unsigned integer format? Give the answer below. It is not necessary to perform a screenshot this time.

If light is declared as:

`light dq 13, -49, 22, -41, 7, 12, 89, -13, 36`

The gdb command would be:

`x/5ug light`

12. We have a big array like `costs resq 150`. The array is full. All arrays reside in memory, and therefore, there is a memory address associated with each number the array costs.

Show how to obtain the memory address of the number stored in array slot with index number 60. Your answer must be one concise instruction that execute quickly.

If the array is declared as:

`costs resq 150`

Each element is 8 bytes so the address of slot 60 is obtained by

`lea rax, [costs + 60*8]`

Conclusion

Time expires at 12:00noon + 5 minutes.

Sign your name and your email address. You must include both data items for the test to be graded. Below is space for the information.

Riley Berry

rberry7@csu.fullertonl.edu

rileyberry7@gmail.com

CWID: 885405613

Change the name of the test file by appending your own name to the existing name. Here is an example:

2025spring240-9midterm-concepts.odt

changes to

2025spring240-9midterm-concepts-Allen-Chan.docx

You may convert this document to doc or docx format, or you may keep it in its original format. All other formats will be rejected.

Attach this document to an email message that will carry the document physically to me. The email carrier is not required to have a message.

Send to holliday@fullerton.edu

Next week classes will resume in person and zoom. The only remaining dates without in-person instruction are test dates.