# [Midterm] Past Exams - 2020 #636

J

Jero Wang STAFF
3 months ago in Exam – Midterm

153
VIEWS

You can find the past exams here: https://cs61c.org/su22/resources/exams/

When posting questions, please reference the semester, exam, and question in this format so it's easier for students and staff to search for similar questions:

**Semester-Exam-Question Number**

For example: **SP20-Final-Q1**, or **SU20-MT2-Q3**

Summer 2020 midterm walkthrough

Fall 2020 midterm Bit Manipulations Walkthrough

Fall 2020 midterm 1 Slip Walkthrough

A

Alexander Ng 3mth #636cf ✓ Resolved

Su20-MT1-Q2b.
Why is 63 bytes of memory not freed and not 62 bytes? Wouldn't 4 * 12 bytes for the nodes created + 13 bytes (for each character in *r) + 1 byte (for null terminator) be 62?

...

P

Peyrin Kao STAFF 3mth #636da

`sizeof(s)` is 14 because it accounts for the null byte, plus 1 is 15.

...

A

Alexander Ng 3mth #636eb

Thanks for replying! I am still confused however, Is it that s is the memory we have not freed? And is there another null byte beyond the null terminator?

My original math is 4*12 bytes per node + (13 + 1) bytes for each character in r + null terminator = 48 + 14 = 62 bytes.

...

P

Peyrin Kao STAFF 3mth #636ec

Replying to Alexander Ng

The calloc call allocates 14+1=15 bytes, so you end up failing to free a block of 15 bytes. The amount allocated by calloc was actually more than we needed to store the string.

...

L Leyla Zokhidova 3mth #636bd ✓ Resolved

```
void free_list(node *n) {
if (n == NULL) return;
node *c = n->nxt;
for (; c != n;){
node *tmp = c; c = c->nxt;
free(tmp);
}
}
```

SU20-MT1-Q2. The solutions say that we are freeing the sentinel node in the stack only and not in the heap. Is it because of the call to calloc  is not accounted for?

•••

P Peyrin Kao STAFF 3mth #636bf

I think this is an error in the solutions, sorry.

•••

T Te-Jung Chen 3mth #636bc ✓ Resolved

In summer 2020 midterm.

Why is this -8192 to 8191? I though in the instructor's lecture he said branch should have an extend zero  is always odd and if we get 2 extra bit it should be [-4096 * 4 to 4094 *4] or [-2 ^ 14, 2 ^ 14 -2]

•••

P Peyrin Kao STAFF 3mth #636be

(edit: see below)

•••

T Te-Jung Chen 3mth #636ca

The answer given is -2^13, 2^13-1

•••

T Te-Jung Chen 3mth #636cb

Is the answer wrong

•••

P Peyrin Kao STAFF 3mth #636cc

Replying to Te-Jung Chen

Sorry, let me explain this in a bit more detail: a standard branch instruction lets you branch by $2^{10}$ instructions in each direction. Since we get two more bits in the immediate, we get 4 times as many instructions to branch to, so that's $2^{12}$ instructions, or $2^{13}$ half-instructions in each direction.

•••

A Alexander Ng 3mth #636cd

Replying to Peyrin Kao

Why does register fields only requiring 4 bits?

•••

T Te-Jung Chen 3mth #636ce

Replying to Peyrin Kao

So you mean they are asking for instructions numbers not address. If it is instruction number should not have negative.

•••

P Peyrin Kao STAFF 3mth #636db

Replying to Te-Jung Chen

Register fields only require 4 bits because the question says there are only 16 registers in your new architecture.

The question is asking about the range of possible offsets, measured in half-word instructions.

•••

T Te-Jung Chen 3mth #636dc

Replying to Te-Jung Chen

How do you know if they are asking instruction numbers or address. In Justin's lecture slide he literally specify the instructions range as address.

•••

P Peyrin Kao STAFF 3mth #636dd

Replying to Te-Jung Chen

The question says: "what is the range of half-word instructions that can be reached?" Branch instructions use relative addressing to reach other instructions, so you wouldn't be able to provide a range of absolute addresses.

•••

T Te-Jung Chen 3mth #636de

Replying to Peyrin Kao

•••

P Peyrin Kao STAFF 3mth #636df

Replying to Te-Jung Chen

Sorry, maybe I'm not fully understanding your question here. The slide is saying that with a 13-bit signed number, you can represent numbers in the range [-4096, 4094]. Then it converts that into 2^10 instructions up/down.

•••

T Te-Jung Chen 3mth #636ea

Replying to Te-Jung Chen

I can't tell if there asking the represent number in range or instructions count. They are both range for instructions just one is address one is number count

Probably just my problem....

•••

P Peyrin Kao STAFF 3mth #636ed

Replying to Te-Jung Chen

Addresses are measured in bytes. The question asks you to measure the range in half-instructions, which are sets of 2 bytes.

•••

Anonymous Alligator 3mth #636ba ✓ Resolved

SP20-Midterm-3b&c

I am confused over why the answer to b:

```
fun->t *= -1; fun-t >>= 1;
```

`printf("%d\n", fun->i[i])` is -1?

and

```
fun->s[0] = '\0';
```

`printf("%d\n", fun->t);` is -256?

Does the original content of the Union get erased when we assigned to a variable of a different type with a larger size or is it retained?

Also, does the size here refer to the integer value size not the type size right?

•••

P Peyrin Kao STAFF 3mth #636bb

Unlike structs, unions store all their components in the same space; we can imagine that while a struct is "Store an array of uints AND an array of ints AND an array of chars AND an int", a union is more "Store an array of uints OR ...". Each write tries to store in a specific part of the union, and ends up modifying the 32 bits of data that *fun is composed of. The reads then end up interpreting the same data in different ways.

•••

Anonymous Starling 3mth #636aa ✓ Resolved

SU20-Midterm-3d

How do you get ABCDD as the first 5 digits of the result?

•••

P Peyrin Kao STAFF 3mth #636ab

auipc will add 0x100 to 0xABCDE000 to get 0xABCDE100. 0xABC will get sign extended to 0xFFFFFABC, which, when summed with 0xABCDE100 will give you 0xABCDDBBC.

•••

Anonymous Sandpiper 3mth #636f ✓ Resolved

SU20-Final-Q11b.ii

For this question, do we not translate the last bit from base three into decimal? Since the mantissa is 2222, with an implicit 1, I think we should translate 1.2222 * 3^15 into decimal as 1 * 3^15 + 2 * 3^14 + 2* 3^13 + 2*3^12 + 2*3^11. However, the solution excludes the 2*3^11 term. Could you explain why we don't use the last bit in translation?

P Peyrin Kao STAFF 3mth #636af

I think there was an error in the solutions here.

Anonymous Starling 3mth #636e ✓ Resolved

SU20-MT1- Q5b

What is one's complement? Is that in scope?

Also, for the second question (2+2 = fish), what is the representation that would result in this being true?

P Peyrin Kao STAFF 3mth #636ac

One's complement is not in scope this summer.

2+2 = fish is not a representation that we've ever talked about. This question is just saying that you could hypothetically come up with a representation where "fish" represents 4 if you wanted to.

Anonymous Starling 3mth #636d ✓ Resolved

SU20-3-viii

Would "addi a0, zero, 0" be the same as "li, a0, 0" in this context?

P Peyrin Kao STAFF 3mth #636ad

Yes, both put the value 0 in register a0.

Anonymous Starling 3mth #636c ✓ Resolved

SU20-1-a

Do you not put "typedef" before the word "union" in nested unions?

P Peyrin Kao STAFF 3mth #636ae

`typedef` is used if you want to give the union another name to refer to it.

Anonymous Sandpiper 3mth #636a ✓ Resolved

SU20-MT1-Q5

For 5.b.A of this question, I am confused with why the answer is true. In my understanding, the range of biased number of n bits is [-2^(n-1) + 1, 2^(n-1)], and the range of two's complement number of n bits is [-2^(n - 1), 2^(n - 1) - 1]. As such, biased number should have one more positive than negative, and two's complement should have one more negative than positive. How can n bits of these two representations cover the same range of integers?

...

P Peyrin Kao **STAFF** 3mth #636b

In bias notation, you can choose the bias to be whatever you want, unless it's something defined in the system (e.g. in floating point). The range you gave is for using a default bias of $-2^{n-1}$, but you could also set the bias to be $-2^{n-1}+1$ which would give you the same range of numbers represented.

...