

Introduction

The goal of this lab is to give you practice with pointers, and structs.

Due: Monday, March 4, 2024 - midnight

Lab Instructions

Consider the following snippet of code:

```
struct NODE{
    int a;
    struct NODE *b;
    struct NODE *c;
};

struct NODE  nodes[5] = {
    {64,  nodes + 2,  NULL},
    {58,  nodes + 4,  nodes + 3},
    {95,  NULL,      nodes + 4},
    {43,  nodes + 1,  nodes},
    {88,  nodes + 3,  nodes + 1}
};

struct NODE *np = nodes + 2;
struct NODE **npp = &nodes[1].b;
```

Task 1:

Using the boxes below you are to draw the representation of the nodes array declared above (including variables and their values). This will help you complete the remainder of this lab.

You are to assume (pretend) the nodes array begins at address location 0x225 and that the computer is 64-bit architecture. This information is needed to help you determine the address of each node and the elements of a node. You will use the information we learned about the size of the struct when calculating the address for each element in the array of structs.

With the above you should have all needed information to complete this diagram.

Address (0x225)	Address_____	Address_____	Address_____	Address_____
a = *b = *c =	a = *b = *c =	a = *b = *c =	a = *b = *c =	a = *b = *c =
nodes [__0__]	nodes [__1__]	nodes [__2__]	nodes [__3__]	nodes [__4__]

Task 2:

You will need to evaluate each expression to determine the value. If the expression cannot be evaluated enter ILLEGAL, if the expression can be evaluated but there is no way to know the value then enter DO NOT KNOW. You should evaluate each expression with the original values shown (In other words do not use the results of one expression to evaluate the next expression.)

Using the above information, complete the following chart.

Expression	Value
nodes	_____
nodes.a	_____
nodes[3].a	_____
nodes[3].c	_____
nodes[3].c->a	_____
*nodes.a	_____
(*nodes).a	_____
nodes->a	_____
nodes[3].b->b	_____

&nodes[3].a	_____
&nodes[3].c	_____
&nodes[3].c->a	_____
&nodes->a	_____
np	_____
np->a	_____
np->c->c->a	_____
npp	_____
npp->a	_____
*npp	_____
*npp->a	_____
(*npp)->a	_____
&np	_____
&np->a	_____
&np->c->c->a	_____

Task 3:

Once you have evaluated each expression you are required to write a program, in C, that will print out the value of each legal expression. Name the “C” file lab6.c.

The format of the output should be the expression, a tab, and the output. The expressions in the blanks above, that produce an address, obviously will not be the same address as the one you print out.

So, you are probably wondering why you don’t just write the program and copy the output to the blanks above. Good question! Some of the expressions above will rely on you knowing what will print in order to even write the print statement. Also, the purpose of this lab is to give you practice with several of the concepts we have recently learned. If you cut corners you will not help you learn. Lastly, you will see some of these or questions like these on a quiz and/or exam.

In your lab6.c file you must have a comment block that has the following information.

```

/*****
*Your name
*Your username
*Lab 6 and your lab section
*****/

```

Submission Instructions:

In the lab 6 module you should find a file called **lab6.json**. Download this file. This is where you will put your answers for tasks 1 and 2. Note the formatting: any pointers you have as answers should begin with “**0x**”, operations that are illegal should be answered with “**ILLEGAL**” in all caps, and unknowable operations should be answered with “**DO NOT KNOW**” in all caps. An example.json file will be provided on Canvas to show what the formatting should look like (this is just an example, none of the values are in any way related to the actual assignment). You will **submit the lab6.json file as well as the lab6.c file from task 3 into gradescope.**

Note: please do not add any comments or headers to the lab6.json file. Just put your answers in, or else the autograder may not be able to read it.