CS111 – Introduction to C Programming
Fall 2025
Programming Assignment #10 (3%)
Due: Friday November 28 (23:59)
C Pointers

Objectives

In this programming assignment, you will continue to learn how to use pointer variables in C through their use in various situations.

Part I: Addresses of C Local Variables (1%)

The C program below contains two functions **`foo1()`** and **`foo2()`**, called by **`main()`**. Each of them contains a local variable (**`x`** and **`y`**). Modified the program so that it prints the values and the addresses of the two local variables. Repeat the experiment by calling **`foo2()`** first and then **`foo1()`** in **`main()`**. Observe the outputs in the two experiments and explain (in the header comment section) how you think C compiler allocates memory for and initialize local variables.

```
#include <stdio.h>

void foo1(int xval) {
   int x;
   x = xval;
   /* print the address and value of x here */
}

void foo2(int dummy) {
    int y;
    /* print the address and value of y here */
}
int main() {
    foo1(7);     // switch the calling order
    foo2(11);    // and run again
    return 0;
}
```

Part II: Pointer Arithmetic (1%)

The following program prints out the value and the address of a character variable before and after it is incremented by 1. Now, modify this program so that it prints out the value, the addresses and the change in the value of a pointer to (a) an integer (**`ip`**) and (b) a double (**`dp`**), before and after the pointer is incremented by 1. The expected output should look like:

```
ip is initially at xxxxxxx.
The integer at ip is xx.
ip+1 is 0xxxxxxxx.
The address in ip has changed by x.
dp is initially at 0xxxxxxxxx.
The double at dp is x.xxxxxx.
dp+1 is 0xxxxxxxx.
The address in dp has changed by x.
```

where "**x**...**xx**" represent values that are dependent on your implementation and your computer.

```
#include <stdio.h>

int main( )
{
    char c = 'Z';
    char *cp = &c;

    printf("cp is %p.\n", cp);
    printf("The character at cp is %c.\n", *cp);

    /* Pointer arithmetic - see what cp + 1 is */
    cp = cp + 1;
    printf("cp is %p.\n", cp);
    return 0;
}
```

Part III: Swapping Function for Strings

Write a C function called **`swap_strs()`** that swaps two strings. **`swap_strs()`** is called by the following **`main()`** , so that **`s2`** gets printed first and then **`s1`**. Do not print in **`swap_strs()`**.

```
int main() {
    char *s1,*s2;

    s1 = "I should print second";
    s2 = "I should print first";
    swap_strs(s1,s2);
    printf("s1 is %s\n", s1);
    printf("s2 is %s\n", s2);
    return 0;
}
```

Marking and Assignment Submission

Once the program is completed, add a header documentation section and inline comments to explain how each program works. The header section should include information about the author and the general information about this assignment as well as about the use of AI as is described in our "AI policy". Each function should have its own header as the main function to describe the input and output the function and the algorithm used to compute output from the input. (This is called "proper documentation" below.) Name the programs for the three parts as **`pa10p1, pa10p2`** and **`pa10p3.c,`** respectively. Submit these three files as well as the zipped folder of the three files, called **`pa10.zip`**, as four separate files through Blackboard. Be prepared to explain to a TA in the next lab session about your solution to this assignment. This programming assignment is worth a total of 3% with 1% for each part.