Exercise 1:
Complete the code below by creating a pointer to the local variable n called ptr_to_n, and use it to increase the value of n by one.

```
#include <stdio.h>

int main(void) {
  int n = 10;
  /* your code goes here */




  /* testing code */
  if (ptr_to_n != &n) return 1;
  if (*ptr_to_n != 11) return 1;

  printf("Done!\n");
  return 0;
}
```

Exercise 2: Consider the following code:

```
  int x = 101;
  int y = 202;
  int* ptr_to_x = &x;
  int* ptr_to_y = &y;

  *ptr_to_x = 1001;
  *ptr_to_y = 2002;
  printf("New value of x = %d\n", *ptr_to_x);
  printf("New value of y = %d\n", *ptr_to_y);

  int firstOp = *ptr_to_x + y;
  int secondOp = x + *ptr_to_y;
  int thirdOp = *ptr_to_x + *ptr_to_y;
  printf("Values of three ops = %d %d %d\n", firstOp, secondOp,
thirdOp);
```

What is the output of this code?

```
New value of x = _____________

New value of y = _____________

Values of 3 ops = _____________    _____________    _____________
```

Exercise 3:
Consider the following code:

```
int my_array[] = {1, 5, 10, 15};
int* array_ptr = my_array;

(*array_ptr)++
printf("Value of *array_ptr: %d\n", *array_ptr);
array_ptr++;
printf("Value of array_ptr: %p\n", array_ptr);
array_ptr++;
*array_ptr = 30;
printf("Value of my_array[2]: %d\n", my_array[2]);

char name[] = "Dr. Summet";
char* char_ptr = name;

printf("Value of *char_ptr: %c\n", *char_ptr);
(*char_ptr)++;
printf("Value of *char_ptr: %c\n", *char_ptr);
char_ptr++;
printf("Value of *char_ptr: %c\n", *char_ptr);
```

Here is a picture showing the data and pointers just after they are declared.

a. Draw a similar picture that shows the state of the pointer and data after all the code executes.

b. If the starting address of `my_array` is 1000 and `name` is 3000, draw a picture of memory after the code executes. Assume that integers take 4 bytes of memory and characters take 1 byte of memory and that memory is "byte addressable" as we discussed last week.

Variable	Memory address	Value

c. Give the output of the code, given the assumptions about addresses in part b.

```
Value of *array_ptr: _________

Value of array_ptr: ________

Value of my_array[2]: _________

Value of *char_ptr: _________

Value of *char_ptr: ________

Value of *char_ptr: ________
```