Name: Inho Choi **ID:** 1801987.

1. Suppose you have the memory given right, where x is an integer and arr is an integer array with 4 elements.

a. Given the following commands what is seen at the console?

cout<<x;	42
cout<<&x;	x9004
cout<<pa;	x9004
cout<<&pa	x9016
cout<<*pa;	42
cout<<pb;	x9048

cout<<&pb;	x9024
cout<<*pb;	-1
cout<<p2;	x9016
cout<<&p2;	x9032
cout<<*p2;	x9004
cout<<**p2;	42

x	42	x9004
		x9008
pa	x9004	x9016
		x9020
pb	x9048	x9024
		x9028
p2	x9016	x9032
		x9040
arr	32	x9044
	-1	x9048
	8	x9052
	12	x9056

b. Without using the identifier x, what is the syntax to overwrite the value currently in x with the value 33?

*pa = 33;

c. Understanding that both pa and pb store addresses to integers, in plain language what do you suppose p2 stores?

p2 stores address value of pointer variable (&pa).

d. Suppose p2 was declared and initialized in a single line resulting in what we see in the memory diagram. What is the syntax for p2's declaration/initialization?

int **p2 = &pa

e. Without using the identifier x or pa, what is the syntax to overwrite the value currently at memory location x9004 with the value 13?

**p2 = 13

f. What is the syntax to assign pb's address to p2?

p2 = &pb.

Name: Inho Choi ID: 180187

2. Consider this implementation with the following usage:

Usage of this function might look like (Note that the **usage is correct** and should not be modified):

```
int count = 0;
compareCstrings("tacocat", "TACOCAT", &count);// should set count to 0
compareCstrings("Harry", "Malfoy", &count); // should set count to 1
compareCstrings("SMC","SBCC", &count);      // should set count to 2


void compareCstrings(const char *str1, const char *str2, int count) {
      count = 0;
      while (str1 != '\0' || str2 != '\0') {
            if (*str1 == *str2)
                  count++;
            str1++;
            str2++;
      }
}
```

Identify the errors in the above implementation and rewrite the function so that it satisfies specification. Try to keep the general form of the original code, you should not have to add or remove any lines of code, just modify the existing ones. There may be issues with one or more of the input parameters. You may not use the indexing operator [].

```
void compareCstrings(const char *str1, const char *str2, int *count)
{
    *count=0;
    while(*str1 != '\0' && *str2 != '\0') {
        if(*str1 == *str2)
        (*count)++;
        str1++;
        str2++;
    }
}
```

Name: Inho Choi **ID:** 1801187

3. Consider the following program which consists of functions: main, foo, and bar. For each of the operations indicate the state of the array at the time of those operations in the boxes to the right of main.

-1 -5

```
void foo(int* a, int *b) {
      int* temp = a;
      a = b;
      b = temp;
}

void bar(int* a, int *b) {
      int temp = *a;
      *a = *b;
      *b = temp;
}

int main() {

   int arr[6]
     = { 13, 6, 3, 17, 80, -5 };
   int *ptr = &arr[2];
        3
   *(ptr + 2)  = 36;

   ptr -= 2;

   ptr[0] = -1;

   *(arr + 1) = 42;

   foo(&arr[0], &arr[5]);

   bar(&ptr[2], arr+2);

}
```

13	6	3	17	80	-5

13	6	3	17	36	-5

13	6	3	17	3	-5

-1	6	3	17	36	-5

-1	42	3	17	36	-5

-1	42	3	17	36	-5

-1	42	3	17	36	-5