Question 1.

Write a function that takes in an address of the first element of a block of ints, and sets all the elements at even indices to 0. For example, if the input block is {5, 6, 7, 8}, it should be change to {0, 6, 0, 8}.
    The function signature should be

```
void set_even_to_zero(int *block, int size)
```

Question 2.

Implement two versions of the string.h function strcat: https://www.programiz.com/c-programming/library-function/string.h/strcat .
    In one version, use the index i to access elements of strings.
    In another version, only use pointer arithmetic.

Question 3.

Implement a version of strcmp recursively. Name the function my_strcmp_rec. See the description of strcmp here: https://www.programiz.com/c-programming/library-function/string.h/strcmp.
    Suppose your code looks as follows:

```
char *s1 = "[...]";
char *s2 = "[...]";
```

    Explain the difference between s1 == s2, *s1 == *s2, and strcmp(s1, s2).

Question 4.

For Project 1, you will need to implement "Binary Search Deluxe" in C.
    In this lab, you will implement it in Python.
    "Binary Search Deluxe" takes in a sorted list of integers and a target, and returns both the first and the last index where target appears.
    For example,

```
binary_search_deluxe([1, 2, 3, 10, 10, 10, 12, 12], 10)
```

    should return [3, 5], since the 10's start at index 3 and end at index 5.
    Make sure that the algorithm runs in $O(\log(n))$ time.
    See next page for a hint.

As a hint, here is `binary_search_deluxe_left` which returns the index of the first occurrence of target in L. Assume that target is in L.:

```
def binary_search_deluxe_left(L, target):
  """Return the index of the first occurrence of target in L.
  Assume that target is in L"""
  left = 0
  right = len(L) - 1
    while left <= right:
      mid = (left + right) // 2
      if L[mid] < target:
        left = mid + 1
      else:
        right = mid - 1
  return left
```

Your task is to implement `binary_search_deluxe_right`, which returns the index of the last occurrence of target in L, in $O(\log(n))$ time. (Note: finding the first occurrence and then looking through the list from there is not $O(\log(n))$ time. One idea: what is the condition that indicates that you found the right-most occurrence of target?)

Question 5.

The C library function `atoi` converts a string to an integer. Write your own version of the function, with the signature

```
int my_atoi(char *str)
```

Hints:

- The function `isdigit` (defined in `<ctype.h>`) can be used to tell if a character is a digit

- You can convert a digit `c` to an integer value using `c-'0'`. For example, `'5'-'0'` is 5 since the digits `'0'`, `'1'`, `'2'`, `'3'`, `'4'`, `'5'` appear in sequence in the ASCII table.