CMPT125, Fall 2021

Homework Assignment 1
Due date: Wednesday, October 1, 2021, 23:59

You need to implement the functions in ***assignment1.c.***
Submit only the ***assignment1.c*** file to Canvas.

Solve all 5 problems in the assignment.

The assignment will be graded automatically.
Make sure that your code compiles without warnings/errors, and returns the required output.

Your code MUST compile in CSIL with the Makefile provided.
If the code does not compile in CSIL the grade on the assignment is 0 (zero).
Even if you can't solve a problem, make sure the file compiles properly.

Warning during compilation will reduce points.
More importantly, they indicate that something is probably wrong with the code.

Your code must be readable, and have reasonable documentation, but not too much.
No need to explain i+=2 with // increase i by 2.

An example of a test file is included.
Your code will be tested using the provided tests as well as additional tests.
You are strongly encouraged to write more tests to check your solution is correct, but you don't have to submit them.

You need to implement the functions in ***assignment1.c.***
If necessary, you may add helper functions to the assignment1.c file,
but you cannot add add main() to
Submit only the ***assignment1.c*** file to Canvas.

Question 1 [15 points].
Write a function that gets two ints `a` and `b`.
If `a>b` the function returns a^3+b^2, and otherwise it returns a^2+b^3.

```
int square_cube(int a, int b);
```

For example:

- `square_cube(1, 2)` *should return* `1+8=9`.
- `square_cube(10, 3)` *should return* `1000+9=1009`.
- `square_cube(2, -1)` *should return* `8+1=9`.
- `square_cube(-2, -1)` *should return* `4-1=3`.

Question 2 [15 points].
Write a function that gets 3 pointers `int* a,int* b,int* c`, and rotates the values in their addresses to the left. That is, `a` gets the value of `b`, `b` gets the value of `c`, and `c` gets the value of `a`.

```
void rotate3(int* a,int* b,int* c);
```

For example,

- *if we have* `int x=1, y=2, z=3`, *then after calling* `rotate3(&x, &y, &z)` *we should have* `x==2, y==3`, *and* `z==1`.
- *if we have* `int x=7, y=1, z=6`, *then after calling* `rotate3(&x, &y, &z)` *we should have* `x==1, y==6`, *and* `z==7`.

Question 3 [20 points].
Implement the function that gets a string `str`, changes all digits of `str` to `0` (zero), and returns the number of digits in the string.

```
int digits_to_zero(char* str);
```

For example:

- *If `str` is `"12ab0"`, then the function should change it to `"00ab0"`, and return 3.*
- *If `str` is `"abcde"`, then the function should keep it as is, and return 0.*

[Hint 1: you can check if a char is a digit using its numerical value. The numerical values of the digits are consecutive. For example, we have ‘3’ + 2 == ‘5’]

[Hint 2: you can also use the function `isdigit(char c)` implemented in the library <ctype.h> https://www.programiz.com/c-programming/library-function/ctype.h/isdigit]

Question 4 [20 points].
Implement the function that gets an array of ints and its length, and returns the maximum of the ***absolute values****.*

```
int max_abs(const int* arr, int len);
```

For example:

- *On input `[1,-3,7,4]` the function should return `7`.*
- *On input `[2,-3,2,-4]` the function should return `4`.*
- *On input `[0,-3,2]` the function should return `3`.*

Question 5 [30 points].
Write a function that gets a string containing a positive integer. The function subtracts 1 from that integer and puts the obtained value in the string.

```
void str_subtract_one(char* num);
```

For example:

- *if before we call* `str_subtract_one(str)` *we have* `str=="1997"`*, then after return* `str` *will be* `"1996"`*.*
- *if before we call* `str_subtract_one(str)` *we have* `str=="12345678987650"`*, then after return* `str` *will be* `"12345678987649"`*.*
- *if before we call* `str_subtract_one(str)` *we have* `str=="100"`*, then after return* `str` *will be* `"99"`*.*

1. *You may assume that the input is always legal, i.e., the string is a positive integer correctly formatted.*
2. *Note that the numbers may be larger than the maximum of* `int` *or* `long`*. That is, you should not try to convert string to int.*