Lab 3

Instructions: Create several basic programs to solve exercises involving arrays, loops, and strings.

Objectives:

- Explore arrays, including strings.
- Explore looping with arrays and strings.

Task 1: Random + 1: Fill an array with 10 random integers. Print the 10 random integers, then print them again with +1 added.

```
/home/user/CIS190/Lab3$ ./Lab3a.out
4   1   10  9   1   1   7   3   5   5
5   2   11  10  2   2   8   4   6   6
/home/user/CIS190/Lab3$ ./Lab3a.out
5   1   5   6   10  5   4   1   8   6
6   2   6   7   11  6   5   2   9   7
/home/user/CIS190/Lab3$ ./Lab3a.out
1   2   2   9   2   2   10  3   2   1
2   3   3   10  3   3   11  4   3   2
```

Figure 1. Example outputs for `Lab3a.c`.

- Create a .c source file named "`Lab3a.c`".
- In `Lab3a.c`:
  - Use `#define` to define constants for the following:
    - The array should contain 10 integers.
    - The randomly generated integers should be between 1 and 10 (inclusive).
  - Declare an integer array.
  - In a loop, fill each spot in the array with a random integer. Print the random integer.
  - In another loop, add 1 to each integer in the array. Print the increased value.

Task 2: Rectangle print: Prompt the user for a number of rows and a number of columns, then print a rectangle of increasing integers following these dimensions.

```
/home/user/CIS190/Lab3$ ./Lab3b.out
How many rows? 3
How many columns? 4
1   2   3   4
5   6   7   8
9   10  11  12
/home/user/CIS190/Lab3$ ./Lab3b.out
How many rows? 5
How many columns? 5
1   2   3   4   5
6   7   8   9   10
11  12  13  14  15
16  17  18  19  20
21  22  23  24  25
```

Figure 2. Example inputs and outputs for Lab3b.c.

- Create a .c source file named "`Lab3b.c`".
- In `Lab3b.c`:
    - Prompt the user for a number of rows and a number of columns.
    - Use nested loops to print the rectangle. The outer loop should handle the rows, while the inner loop should handle the individual numbers within each row.

Task 3: Word print: Prompt the user for a string, then print each word of that string on a separate line.

```
/home/user/CIS190/Lab3$ ./Lab3c.out
Enter a string: first second third fourth
first
second
third
fourth
/home/user/CIS190/Lab3$ ./Lab3c.out
Enter a string: this is a test string
this
is
a
test
string
```

Figure 3. Example inputs and outputs for `Lab3c.c`.

- Create a .c source file named "`Lab3c.c`".
- In `Lab3c.c`:
    - Prompt the user for a string containing more than one word separated by spaces.
        - `fgets(buffer, MAX_SIZE, stdin)` can be used to obtain string input from the user, where `buffer` is a char array of length `MAX_SIZE`.
    - Loop through the characters in the string, printing the characters one-by-one.
        - When the current character is a space (`' '`), print a `\n` to go to the next line.

Submission details:

- Upload a compressed archive (e.g., .zip) containing `Lab3a.c`, `Lab3b.c`, and `Lab3c.c`.
- The archive should be named `Lab3_LastName`, where `LastName` is your last name.
- If you're on Linux, you can use the following command to create a .tar.gz archive from the terminal:

  ```
  $ tar -czvf Lab3_LastName.tar.gz Lab3a.c Lab3b.c Lab3c.c
  ```

  where `LastName` is your last name.