CENG209: Fall 2022

Homework 2: **Arrays and Functions**

Assigned: Friday, December 2, 2022

Deadline: Wednesday, December 21, 2022

The homework must be completed by Wednesday, December 21. You don’t need to submit the homework. Our TA Elif ŞANLIALP will schedule demos and grade your work.

**Problem 1.** In this program, you will simulate very large POSITIVE integers with arrays. Digits are stored in an array starting with the least significant at index 0. For example, an integer 123 is stored with 3 at index 0, 2 at index 1 and 1 at index 2. You may assume that the integers to be added or multiplied have at most 20 digits. Write the following functions:

* **int add**(**int x**[]**, int sizex, int y**[]**, int sizey, int res**[]), which takes two arrays x and y and their respective sizes, performs addition and stores the result into the array res. Return the number of digits of the resulting sum (i.e. number of digits stored into res).
* **int multiply**(**int x**[]**, int sizex, int y**[]**, int sizey, int res**[]), which takes two arrays x and y and their respective sizes, performs multiplication and stores the result into the array res. Return the number of digits of the resulting product (i.e. number of digits stored into res).
* **int divide**(**int x**[]**, int sizex, int y**[]**, int sizey, int res**[]), which takes two arrays x and y and their respective sizes, performs integer division of x by y, and stores the result into the array res. Return the number of digits of the result.
* **int mod**(**int x**[]**, int sizex, int y**[]**, int sizey, int res**[]), which takes two arrays x and y and their respective sizes, performs of x mod y, and stores the result into the array res. Return the number of digits of the result.
* **printNumber**(**int x**[]**, int sizex**), which prints the integer represented by array x.
* a **main**() function that creates some numbers and tests the add, multiply, division and mod operations.

**Problem 2.** Imagine a mouse that walks within a 50 by 50 array. The mouse holds a pen in one of two positions: up or down. While the pen is up, the mouse does not write anything. While the pen is down, the mouse marks each array location it passes by. You will read a sequence of commands (one per line) from standard input (I suggest using an input text file and use input redirection) and walk the mouse. The mouse always starts at location (0,0) with its pen up. Possible commands are:

* 1 Pen up
* 2 Pen down
* 3 Turn right
* 4 Turn left
* 5,n Move forward n spaces
* 6 Print the array, putting an asterisk for marked locations, putting a space for unmarked locations. However, put one extra space character between consecutive columns.
* 9 End of data marker.

For example, the following set of commands should draw a 10 by 10 array as:

\* \* \* \* \* \* \* \* \* \*

\* \*

\* \*

\* \*

\* \*

\* \*

\* \*

\* \*

\* \*

\* \* \* \* \* \* \* \* \* \*

2

5,10

3

5,10

3

5,10

3

5,10

1

6

9

Provide at least three input text files to draw interesting shapes. Make sure to check for cases when the mouse tries to move outside the array, and do not let it move beyond array boundaries.