

CS 100
Lab #7

1. Write a function **getDigit** that takes two integers, **n** and **k**, and returns the **k**th digit (from the right hand side) in **n**. Use this method to print out the result for a given **n** and **k** that are to be taken as inputs from the user. If **n** does not have **k** digits, the function returns 0.

2. Write a function **gcd** that takes two integers, **a** and **b** from the user, and then finds the greatest common divisor of **a** and **b**. For example, the gcd of 12 and 32 is 4.

3. Write a method called **minimumDigit** that takes an integer **n** and returns an array of length 2 such that the first element of the array is the minimum digit and the second element is the position of the digit from the left hand side. If the minimum value is seen more than once then your function should return the first position it has ever occurred.

Examples:

- a. For **n** = 123, the method returns [1, 3].
- b. For **n** = 42954, the method returns [2, 4].
- c. For **n** = 6661093123, the method returns [0, 6].

4. What is the output of this code? Firstly, try to find the answer by hand and then use Matlab to check your answers.

```
res = 0;
x = 32;
for j = 15:-3:3
    if (j > 6)
        res = res - 2 * j;
    elseif (j == 8)
        fprintf( ' this is the breaking point\n');
        break;
```

```

        else
            res = res + j;
        end
        x = x / 2;
        fprintf( ' the value of x is %d\n ' , x);
        fprintf( ' the value of res is %d\n ' , res);
        fprintf( ' the value of j is %d\n ' , j);
    end
end

```

5. Write a program that checks whether the elements in an array A, are sorted in ascending order.

Examples:

- a. For A = [1,2,3,4], your program outputs "A is sorted".
- b. For A = [-1, -2, -3, -4, -5], your program outputs "A is not sorted".
- c. For A = [1, 2, 2, 2, 2], your program outputs "A is sorted".
- d. For A = [-1, 0, 1, 4, 55, 90] your program outputs "A is sorted".

6. Write a code to generate the NxN matrix A given as below:

1	2	3	...	N-1	N
2	3	4	...	N	N+1
3	4	5	...	N+1	N+2
:	:	:	...	:	:
N-1	N	N+1	...	2N-3	2N-2
N	N+1	N+2	...	2N-2	2N-1