Northern Michigan University (Marquette Co, MI)

CS344-01-24F: iOS / iPhone Programming (Andrew A. Poe) Name: ________________________________
Midterm Examination (Exam 1) Page 1/2 Monday 14 October 2024 1:00 P.M. EDT

Time: 50 minutes.

1. Write the following method:

```
func CountMe (_ s:String) -> Int { ... }
```

This function takes the string s and returns the number of times "POE" appears as a substring within it. For example, if s is "EDGAR ALLAN POE WROTE POETRY" the method should return 2. Do not worry about lowercase. ONLY search for uppercase examples of POE.

You might find this useful:

```
let c:[Character] = Array (s)
```

copies string s into an array of character.

```
func CountMe (_ s:String) -> Int {

    let c:[Character] = Array (s)
    var ct:Int = 0
    for i in 0..<c.count-2 {
        if c[i]=="P" && c[i+1]=="O" && c[i+2]=="E" {
            ct += 1
        }
    }
    return ct
}
```

2. Write the following method:

```
func NegateDiagonals  (_ M:[[Int]])->[[Int]]
```

This method returns a matrix identical to M except that the entries along the diagonals are negated. For example, if the matrix were

1	2	3		-1	2	-3
4	5	6	the returned matrix would be	4	-5	6
7	8	9		-7	8	-9

(Notice that even though 5 is on both diagonals, it is only negated once.)

Reminder: you cannot change the matrix M; Swift will not allow it. You will have to create a new matrix.

```
func NegateDiagonals (_ M:[[Int]]) -> [[Int]] {

    var Z:[[Int]] = []
    for i in 0..<M.count {
        var row:[Int] = []
        for j in 0..<M.count {
            if (i==j || i+j == M.count-1) {
                row.append (-M[i][j])
            } else {
                row.append (M[i][j])
            }
        }
        Z.append(row)
    }
    return Z
}
```

3. Write the methods in ViewController that display two TextFields on the screen (tf1 and tf2) and one Button (butt) (all of these variables must be declared but they have been placed on the Storyboard). The button ALREADY contains the text "CLICK" (from the Storyboard). The ViewController should manually add an action to the button so that when the button is clicked, tf2 will contain the number Q's (upper or lowercase) appearing in tf1. For example, if tf1 contains "Queen Elizabeth quietly quaffed her Nestle Quik", tf2 should contain "4" when the button is clicked. Note: how do you put an integer into a text field?

```
class ViewController: UIViewController {

    @IBOutlet var tf1:UITextField?
    @IBOutlet var tf2:UITextField?
    @IBOutlet var butt:UIButton?
    override func viewDidLoad() {
        super.viewDidLoad()
        butt!.addTarget(self, action: #selector(Click), for:
UIControl.Event.touchUpInside)
    }

    @objc func Click () {

        var ct = 0
        for ch in tf1!.text! {
            if ch=="Q" || ch=="q" {
                ct+=1
            }
        }
        tf2!.text = "\(ct)"
    }
}
```

4. Write a ViewController that does not use the storyboard and creates one label. The label should be 50 units high; it should take up half the width of the view, and should be located at the very lower right corner of the view (but none of the label should be hanging off the side or the bottom). The text of the label should read "I rock at midterms."

```
class ViewController: UIViewController {

    var lab:UILabel?
    override func viewDidLoad() {
        super.viewDidLoad()
        lab = UILabel()
        lab!.frame = CGRect (x:view.frame.size.width/2 , y:
view.frame.size.height-50, width: view.frame.size.width/2, height: 50)
        lab!.text = "I rock at midterms."
        view!.addSubview (lab!)
    }
}
```