

1	2	3	4	5	6	7	8	9	10
0	1	2	3	4	5	6	7	8	9
5	15	2	25	4	17	85	22	33	48

```

int findMax(array) {
    int max = 0;
    for (i = 0; i < 10; i++) {
        if (array[i] > array[max]) {
            max = i;
        }
    }
    return max;
}

```

$O(2n)$   
 ~~$O(n)$~~

max    i  
 0        0  
 1        1  
 3        2  
 6        3  
          4  
          5  
          6  
          7  
          8  
          9

	0	1	2
0	4	22	15
1	75	83	17
2	92	105	98

$O(9)$   
 $O(n * n)$   
 $O(n^2)$

```

int Max (array) {
    int max i = 0;
    int max j = 0;
    for (i=0; i<3; i++) {
        for (j=0; j<3; j++) {
            if (array[i][j] < array[max i][max j]) {
                max i = i;
                max j = j;
            }
        }
    }
    return array [max i][max j];
}
  
```