

- Viết hàm phát sinh mảng 2 chiều các số nguyên.

Mảng 2 chiều
Các phần tử trong mảng 2 chiều:
<ul style="list-style-type: none"><li>[5,12,17,9,3]</li><li>[13,4,8,14,1]</li><li>[9,6,3,17,21]</li></ul>

Hướng dẫn:

#### App.component.html

```
<div class="container" >
  <div class="row">
    <div class="col-sm-5">
      <div class="card">
        <div class="card-header text-center text-info">
          <h5>Mảng 2 chiều</h5>
        </div>
        <div class="card-body">
          <label for="">Các phần tử trong mảng 2 chiều:</label>
          <div class="text-danger text-justify">
            @for (item of Array ; track $index) {
              <ul>
                <li>{{item}}</li>
              </ul>
            }
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

#### App.component.ts

```
import { Component } from '@angular/core';
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'Lab3';
  Array: any[] = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
}
```

2. Viết hàm phát sinh mảng 2 chiều ngẫu nhiên các số nguyên.

**Mảng ngẫu nhiên 2 chiều**

Nhập số dòng:

Nhập số cột:

Ngẫu nhiên

Các phần tử ngẫu nhiên trong mảng 2 chiều:
 

- [9,8,5]
- [6,2,1]
- [3,5,8]

Hướng dẫn: Hàm tạo mảng 2 chiều ngẫu nhiên

`bai2.component.ts`

```
import { Component } from '@angular/core';
```

```
@Component({
  selector: 'app-bai2',
  templateUrl: './bai2.component.html',
  styles: [
  ]
})
export class Bai2Component {
  R: number = 0
  C: number = 0
  Array: any[] = [];
  addToArray(r: number, c: number) {
    this.Array = []
    for (let i = 0; i < r; i++) {
      this.Array[i] = [];
      for (let j = 0; j < c; j++) {
        this.Array[i][j] = Math.floor(Math.random() * 20) + 1
      }
    }
    console.log(this.Array)
  }
}
```

### Tính Tổng

3. Viết hàm tính tổng các phần tử trên cùng một dòng.

**Mảng ngẫu nhiên 2 chiều**

Nhập số dòng:  Nhập số cột:  Ngẫu nhiên

Các phần tử ngẫu nhiên trong mảng 2 chiều:

- [6,9,7,4]
- [10,1,7,1]
- [6,3,5,4]
- [6,8,8,7]

Tính tổng các phần tử trên cùng một dòng:

**26 19 18 29**

Hướng dẫn: Hàm tính tổng dòng

```
SumRows = () => {
  var str = '', sum
  for (let i = 0; i < this.Array.length; i++) {
    sum = 0
    for (let j = 0; j < this.Array[i].length; j++) {
      sum += this.Array[i][j]
    }
    str += sum + " "
  }
  return str
}
```

4. Viết hàm tính tổng các phần tử trên cùng một cột.

**Mảng ngẫu nhiên 2 chiều**

Nhập số dòng:  Nhập số cột:  Ngẫu nhiên

Các phần tử ngẫu nhiên trong mảng 2 chiều:

- [10,3,4,2,10,7]
- [8,4,10,10,8,10]
- [2,2,6,4,2,1]

Tính tổng các phần tử trên cùng một cột:

**20 9 20 16 20 18**

Hướng dẫn: Hàm tính tổng cột

```
SumCols = (r: number, c: number) => {
  var str = '', sum

  for (let i = 0; i < c; i++) {
```

```

    sum = 0
    for (let j = 0; j < r; j++) {
        sum += this.Array[j][i]
    }
    str += sum + " "
}
//console.log(str)
return str
}

```

5. Tổng hợp của bài 3 & bài 4.

### Mảng ngẫu nhiên 2 chiều

Nhập số dòng:

Nhập số cột:

Ngẫu nhiên

Các phần tử ngẫu nhiên trong mảng 2 chiều:

- [5,7,4,10,10,9]
- [2,2,3,1,4,10]
- [6,4,5,6,3,5]

Tổng dòng: ☒ Tổng cột: ☒

Tính tổng các phần tử trên cùng một cột:

13 13 12 17 17 24

Tính tổng các phần tử trên cùng một dòng:

45 22 29

6. Viết hàm tính tổng các phần tử nằm trên đường chéo chính của ma trận vuông.  
\$i==\$j;

```

Cheochinh(r: number, c: number) {
    var sum = 0
    //console.log('hi')
    for (let i = 0; i < r; i++) {
        for (let j = 0; j < c; j++) {
            if (i == j)
                sum += this.Array[i][j]
        }
    }
    //console.log(str)
    return sum
}

```

7. Xây dựng trang web kết quả học tập sinh viên.



## Information Student ITC

Full name: Nguyễn Thị Tèo

Birthday: 02/06/2000

Gender: Nam

mark: 8.5

### Hướng dẫn:

- Dùng lệnh ng generate interface students để xây dựng cấu trúc dữ liệu đối tượng
- Xây dựng cấu trúc như sau:

```
export interface Student {  
  fullname: string,  
  birthday: Date,  
  gender: string,  
  mark: Number,  
  pic: string  
}
```