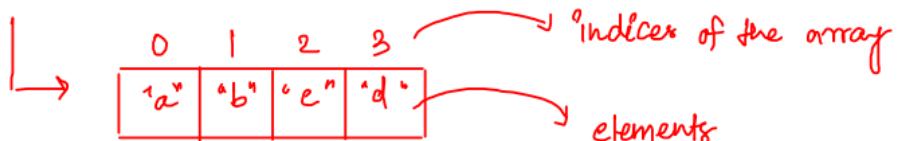


① const friends = ["a", "b", "c", "d"] ;



diagrammatic representation
(this is how it looks like)
in the memory

how many elements, $n = 4$.

② How to access individual elements ?

friends[0] → "a"

friends[1] → "b"

④ If there are 'n' elements in the array, then find last element ?



③ No. of elements ?

friends.length → 4

e.g; friends[n-1]

→ friends[4-1]

→ friends[3] ⇒ "d"

arr[n-1]

⑤ If there are 'n' elements, how to print each ele on new line?

$$\text{Const air} = \begin{array}{ccccccccc|c} 0 & 1 & 2 & \dots & \dots & n-3 & n-2 & n-1 \\ \boxed{} & \boxed{} & \boxed{} & & \ddots & & & & \boxed{} \end{array}$$

$$n=4$$
$$\text{id}_x = \cancel{x} \times \cancel{x} \times \cancel{x} \times \cancel{x}$$

1000 elements

1000
console.
log

↓

console.log (arr[0]);
console.log (arr[1]);
console.log (arr[2]);
:
:
:
console.log (arr[n-2]);
console.log (arr[n-3]);

Use a
loop

```
* Iterate on index  
for(let idx = 0; idx < n; idx++) {  
    console.log(`arr[${idx}]`);  
}
```

- ① $\int dx = 0 \rightarrow \text{an } [0]$
 - ② $\int dx = 1 \rightarrow \text{an } [1]$
 - ③ $\int dx = 2 \rightarrow \text{an } [2]$

\vdots

 - ④ $\int dx = n-1 \rightarrow \text{an } [n-1]$

⑥ Can I change any element in the array?

const arr = [⁰a, ¹b, ²c, ³d]

arr[1] = "e" \Rightarrow [⁰a, ¹e, ²c, ³d]

* JS Arrays are mutable, that means
the elements can be changed.

⑦ How to add new elements to the array?

const arr = [⁰a, ¹b, ²c, ³d, ⁴e]

+ arr.push("e");

elements added
at the end.

⑧ arr.pop();

cl(arr) → [a, b, c, d]

7:20 - 7:35 break

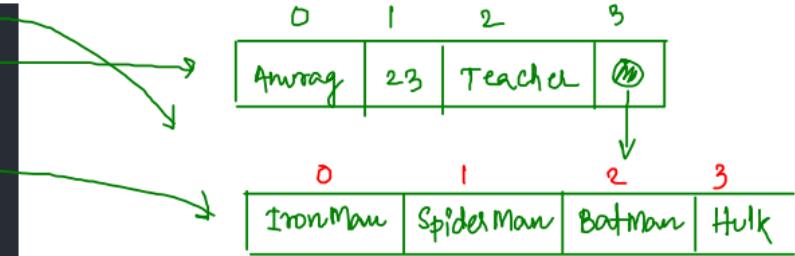
⑨

```
717 const myArr = [firstName, age, job, friends];
718 console.log(myArr);
719 console.log(myArr[0]);
720 → Anurag
721 console.log(myArr[1]);
722 → 23
723 console.log(myArr[2]);
724 → Teacher
725
726 // given myArr, find the number of friends ?
727 // we need to find the length of friends
728 console.log(myArr[3].length);
729
730 // given myArr, print the best friend,
731 // if friends are ordered from best to worst ?
732 console.log(myArr[3][0]);
733
734 // given myArr, print 2nd indexed friend
735 console.log(myArr[3][2]);
736
737 // given myArr, print last indexed friend
738 const lastIdx = myArr[3].length - 1;
739 console.log(myArr[3][lastIdx]);
```

$$\text{lastIdx} = 4 - 1 = 3$$

→ myArr[3][lastIdx]

myArr[3][3] → Hulk

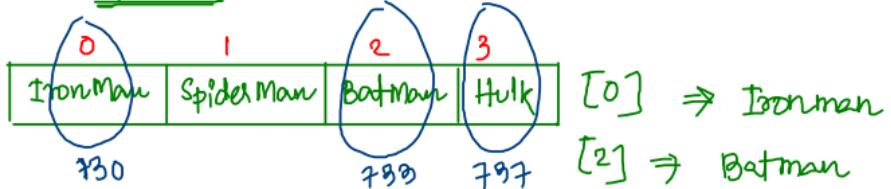


myArr[3].length



⇒ ④ (No. of friends)

myArr[3][0]



* Find the Maximum Ele and its Index :

Ex: arr :

0	1	2	3	4
1	2	7	4	10

Op: 10, 4

Notes

shop	choc
0	1
1	2
2	7
4	10

↓
Idx

↓
ele

1. go to shop 0 and note down

2. compare shop 1 and notes

→ Shop 1 is selling more choc compared the shop in my notes

1st class
what is Algorithm?
- steps

shop 0 1 2 3 4

1	2	7	4	10
---	---	---	---	----

Kid



Chocolate street

→ 100 shops

the kid should find the shop which sell maximum chocolates

* Kid wants to buy many chocolates but he can only buy from one shop.

* minEle = ∞ *Smallest value possible*
let maxEle = $-\infty$ } initial values
let maxEleIdx = -1 *any num that cannot be any*
for(let i = 0; i < n; i++) { *Index*

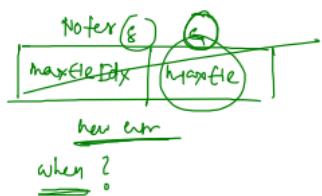
if(arr[i] > maxEle) {

 maxEle = arr[i];

 maxEleIdx = i;

}

}



if (numChocShop > chocInNoter) {
 charge Noter

}

Eg: [1, 2, 7, 4, 10]
↓ ↓ ↓ ↓ ↓
0 1 2 3 4
X > -∞

maxEle = $-\infty$ ~~1~~ ✗ ~~7~~ ✗ ~~4~~ ~~10~~ ans
maxEleIdx = -1 ~~0~~ ✗ ~~2~~ ✗ ~~4~~ ~~5~~ =

① i=0
→ 1 > $-\infty$ ✓

② i=1
→ 2 > 1 ✓

③ i=4
→ 10 > 7 ✓

④ i=2
→ 7 > 2 ✓

⑤ i=5 ($i < 5$) ✗

⑥ i=3
→ 4 > 7 ✗

* Max diff b/w any 2 elements :

eg: $[16, 24, 89, 35]$

$$\begin{aligned} & \text{16-24} \\ &= 8 \\ & (16, 24) = 8 \quad (24, 89) = 65 \quad (89, 35) = 54 \\ & (16, 89) = 73 \quad (24, 35) = 11 \\ & (16, 35) = 19 \end{aligned}$$

Op: 73

* How to generate all the pairs?

$(0, 1)$ $(1, 2)$ $(2, 3)$ (i, j)
 $(0, 2)$ $(1, 3)$ $i=2$
 $(0, 3)$ $j=1$ $j=3$ starting
 $i=0$ $j=2, 3$ $\text{val} = i + j$
 $j=1, 2, 3$

fix i
for (let $i=0; i < n; i++$) {
 move j

 for (let $j=i+1; j < n; j++$) {

$c1(i, j, \text{val}(i), \text{val}(j));$

}