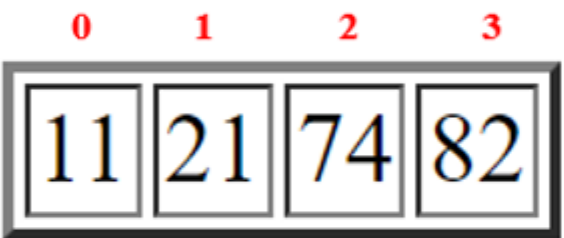


## ***Багатовимірні масиви***

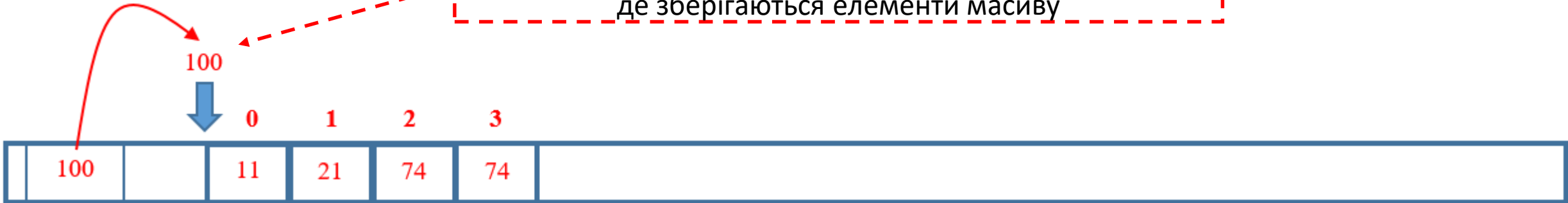
***Вимірність*** – кількість індексів, які необхідно вказати для однозначної ідентифікації елемента масиву

# Одновимірні масиви (для ідентифікації елемента достатньо одного індекса)

```
let arr = [11, 21, 74, 82]
```



Умовна адреса масиву (адреса початку ділянки пам'яті, де зберігаються елементи масиву)

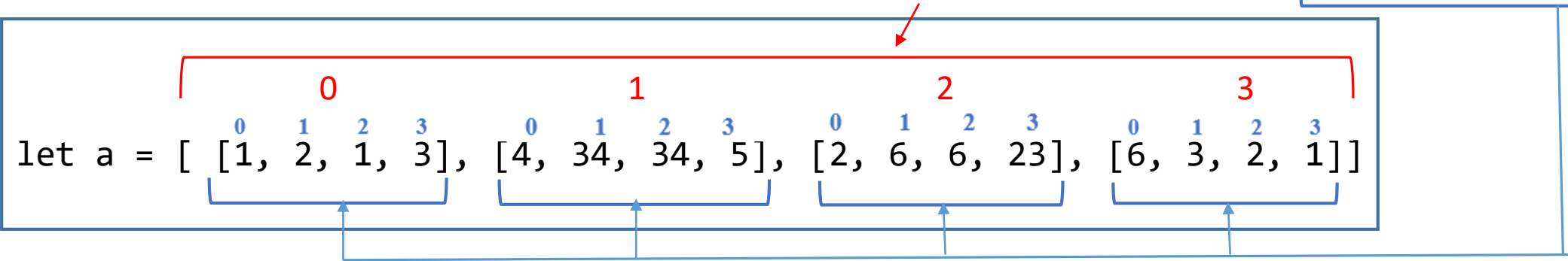


arr  
Адреса масиву

Область пам'яті, де зберігаються елементи масиву

# Двовимірні масиви (для ідентифікації елементів потрібно *два індекси*)

За своєю природою двовимірний масив є одновимірним масивом – з елементами одновимірними масивами рядками



Опис: `let назва_масиву = [ [...список_елементів_рядка 0...], [...список_елементів_рядка 1...], ... ]`

Звертання до елемента: `назва_масиву [номер_рядка][номер стовпця]`

*[ номери рядків ]*

*[ номери стовпців ]*

	0	1	2	3
0	<code>a[0][0]</code>	<code>a[0][1]</code>	<code>a[0][2]</code>	<code>a[0][3]</code>
1	<code>a[1][0]</code>	<code>a[1][1]</code>	<code>a[1][2]</code>	<code>a[1][3]</code>
2	<code>a[2][0]</code>	<code>a[2][1]</code>	<code>a[2][2]</code>	<code>a[2][3]</code>
3	<code>a[3][0]</code>	<code>a[3][1]</code>	<code>a[3][2]</code>	<code>a[3][3]</code>

**a =**

**a =**

	0	1	2	3
0	1	2	1	3
1	4	34	34	5
2	2	6	6	23
3	6	3	2	1

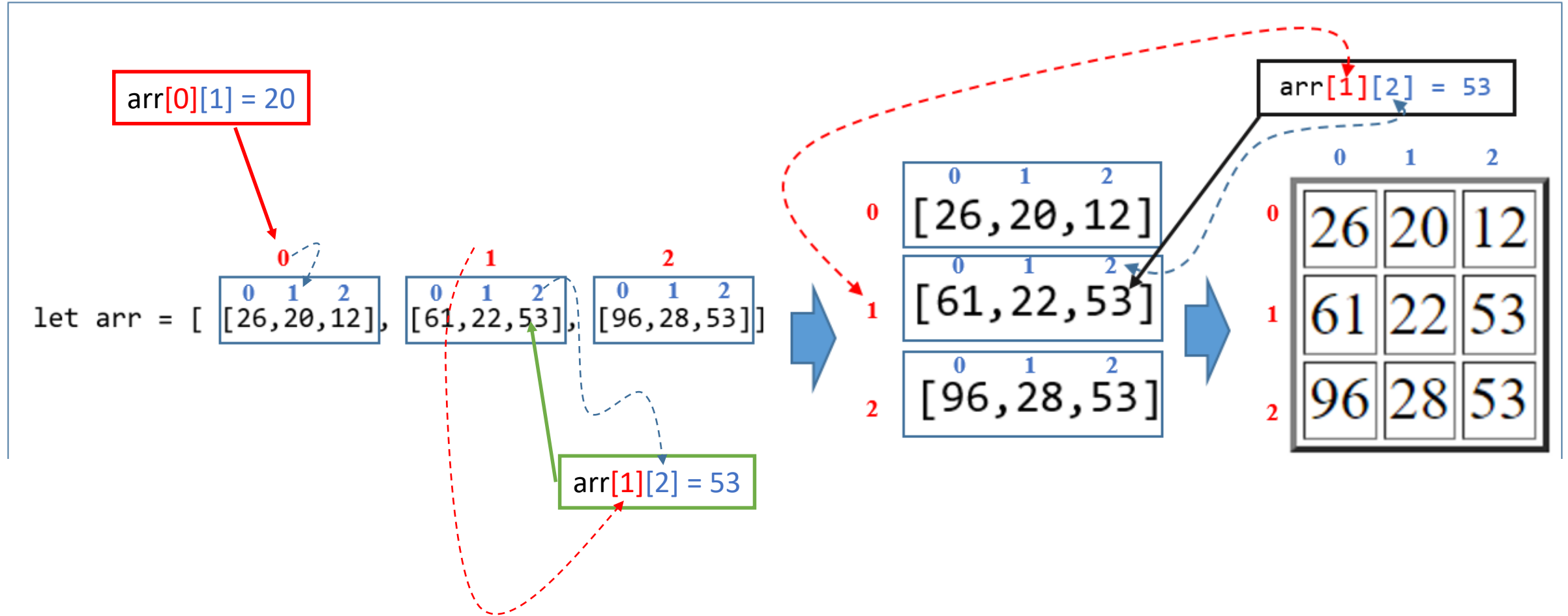
`a[1][2] = 34`

`a[0][3] = 3`

`a[2][3] = 23`

# Двовимірні масиви (для ідентифікації елемента потрібно два індекси)

```
let arr = [ [26,20,12], [61,22,53], [96,28,53] ]
```



# Двовимірні масиви (для ідентифікації елемента потрібно два індекси)

```
let arr = [ [26,20,12], [61,22,53], [96,28,53] ]
```

```
let arr = [ 

|    |    |    |
|----|----|----|
| 0  | 1  | 2  |
| 26 | 20 | 12 |

, 

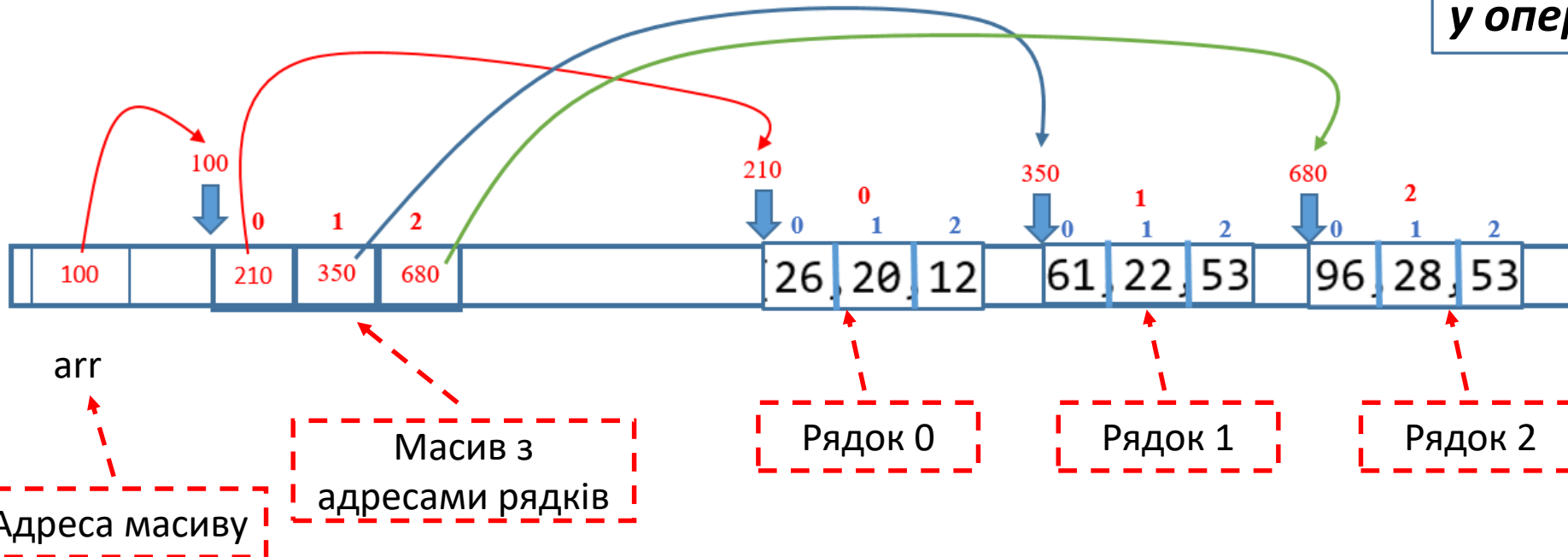
|    |    |    |
|----|----|----|
| 0  | 1  | 2  |
| 61 | 22 | 53 |

, 

|    |    |    |
|----|----|----|
| 0  | 1  | 2  |
| 96 | 28 | 53 |

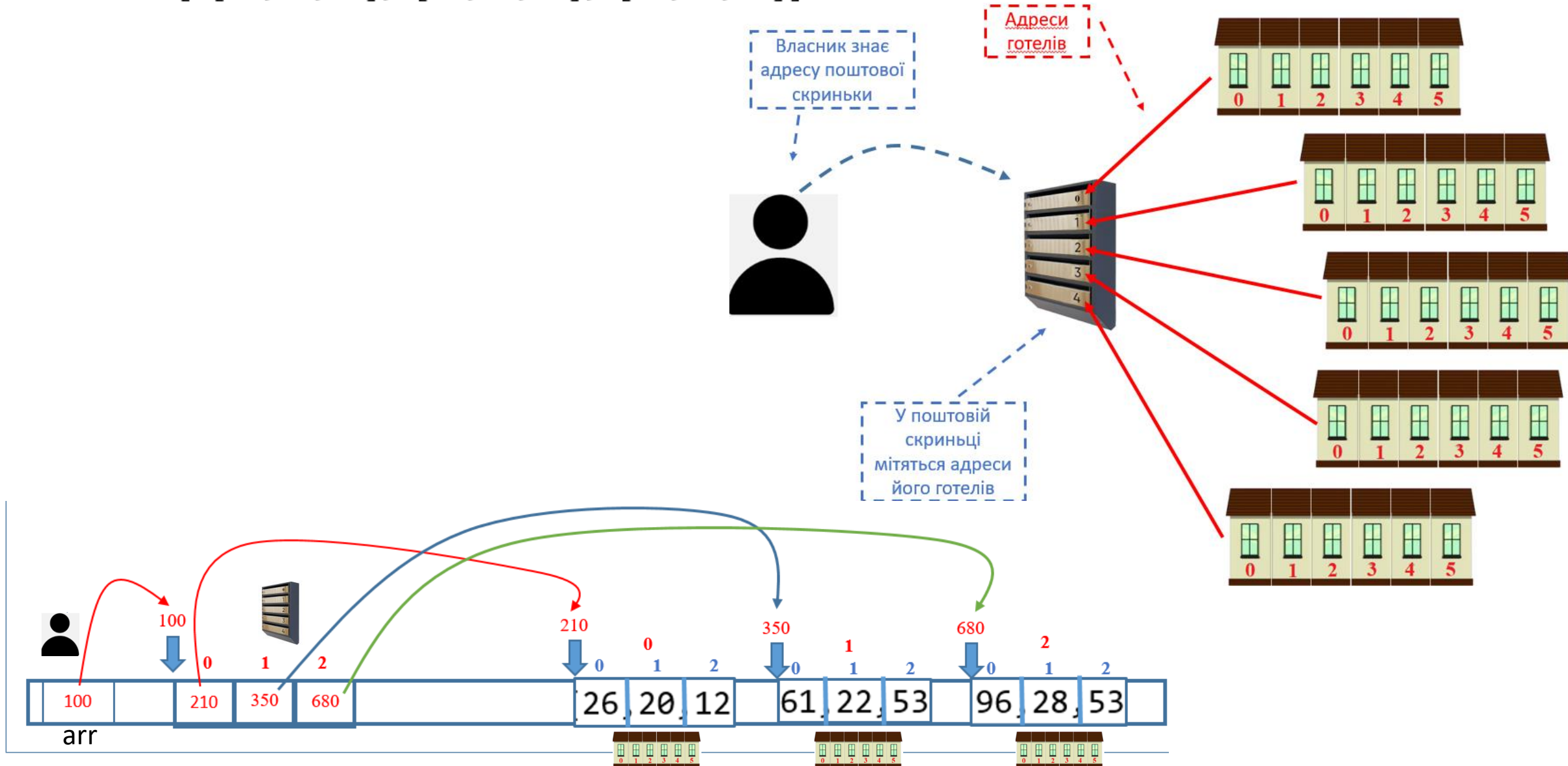
 ]
```

*Схематичне представлення  
у оперативній пам'яті*



# Двовимірні масиви (для ідентифікації елемента потрібно два індекси)

```
let arr = [ [26,20,12], [61,22,53], [96,28,53] ]
```



При перегляді елементів двовимірного масиву один із варіантів полягає у використанні циклів for, при цьому необхідно вказати *як змінюються номери рядків* і *як змінюються номери стовпців*

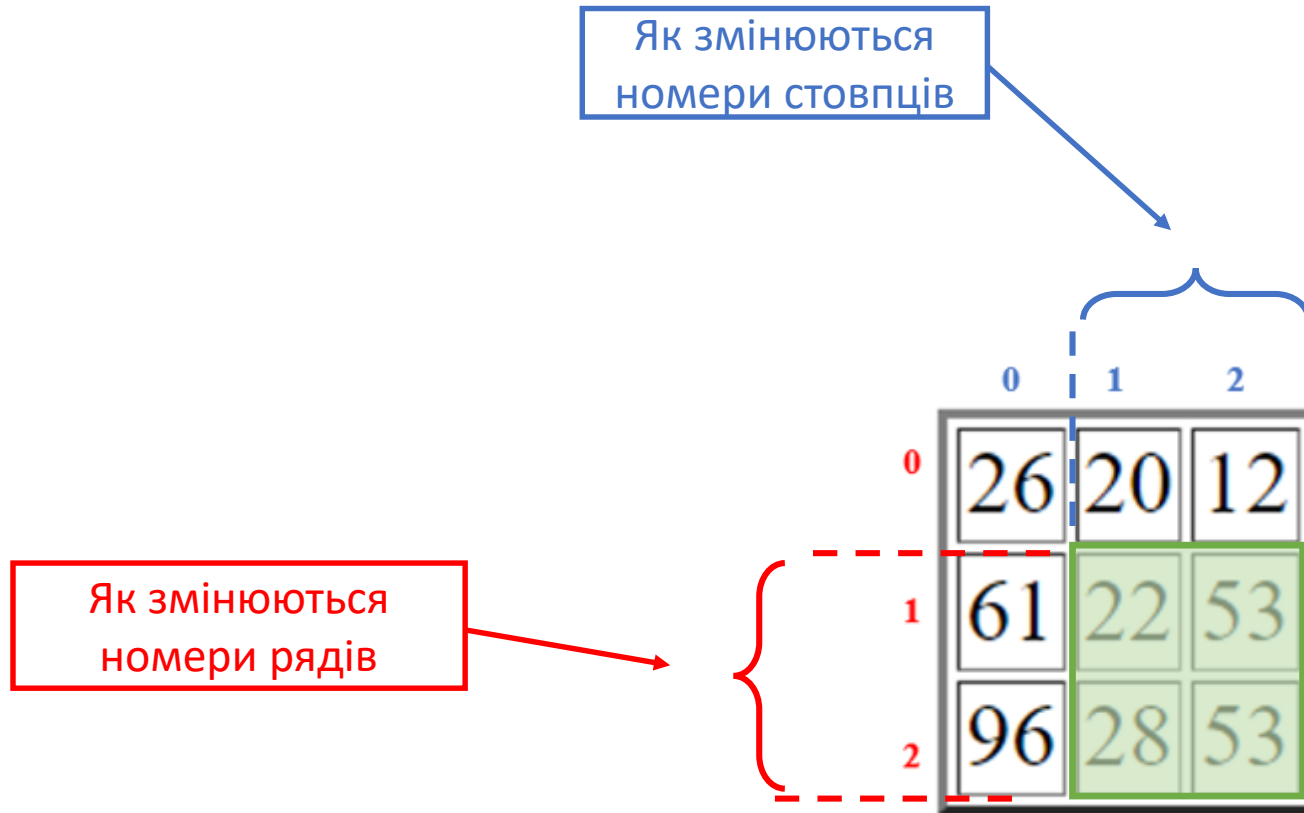
```
for(номер_рядка= ПОЧ.ряд. ; номер_рядка <= КІНЦ.ряд. ; номер_рядка += крок ) {  
  
    for(номер_стовпця = ПОЧ.СТОВП. ; номер_стовпця <= КІНЦ.СТОВП. ; номер_стовпця += крок ){  
  
        .....  
        .... операції з елементом : масив [номер_рядка] [номер_стовпця] .....  
        .....  
    }  
  
}
```

Знайти суму вказаних елементів масиву

	0	1	2
0	26	20	12
1	61	22	53
2	96	28	53



# Знайти суму вказаних елементів масиву



**for**(номер\_рядка=**поч.ряд.**; номер\_рядка <= **кінц.ряд.**; номер\_рядка += **крок**) {

..... що робити з кожним рядком .....

}

```
let arr = [ [26,20,12], [61,22,53], [96,28,53] ]  
let sum = 0
```

```
for (let rowIndex = 1; rowIndex <= 2; rowIndex++) {
```

..... Що робити з кожним рядком .....

```
}
```

Як змінюються  
номери рядів

	0	1	2
0	26	20	12
1	61	22	53
2	96	28	53

**for**(номер\_рядка=**поч.ряд.**; номер\_рядка <= **кінц.ряд.**; номер\_рядка += **крок**) {

**for**(номер\_стовпця = **поч.стовп.**; номер\_стовпця <= **кінц.стовп.**; номер\_стовпця += **крок**) {  
.....  
.... операції з елементом : масив **номер\_рядка** **номер\_стовпця** .....  
.....  
}

}

```
let arr = [ [26,20,12], [61,22,53], [96,28,53] ]  
let sum = 0
```

```
for (let rowIndex = 1; rowIndex <= 2; rowIndex++) {  
  for (let colIndex = 1; colIndex <= 2; colIndex++) {  
    sum += array[rowIndex][colIndex];  
  }  
}  
alert(sum)
```

Як змінюються  
номери стовпців

Як змінюються  
номери рядів

	0	1	2
0	26	20	12
1	61	22	53
2	96	28	53

	0	1	2	3
0	1	2	1	3
1	4	34	34	5
2	2	6	6	23
3	6	3	2	1

Як змінюються  
номери стовпців

	0	1	2	3
0	1	2	1	3
1	4	34	34	5
2	2	6	6	23
3	6	3	2	1

Як змінюються  
номери рядів

**for**(номер\_рядка=**поч.ряд.**; номер\_рядка <= **кінц.ряд.**; номер\_рядка += **крок**) {

**for**(номер\_стовпця = **поч.стовп.**; номер\_стовпця <= **кінц.ряд.**; номер\_стовпця += **крок**) {

.... операції з елементом : масив **номер\_рядка** **номер\_стовпця** .....

}

}

```
let sum = 0
```

```
for (let rowIndex = 1; rowIndex <= 2; rowIndex++) {
```

```
  for (let colIndex = 1; colIndex <= 3; colIndex++) {
```

```
    sum += array[rowIndex][colIndex];
```

```
  }
```

```
}
```

```
alert(sum)
```

Як змінюються  
номери стовпців

Як змінюються  
номери рядів

	0	1	2	3
0	1	2	1	3
1	4	34	34	5
2	2	6	6	23
3	6	3	2	1

Задача. Дано прибуток магазину за K тижнів. Знайти загальний прибуток

Задача. Дано прибуток магазину за K тижнів. Знайти загальний прибуток

```
function getRandomNumber(minValue, maxValue) {  
  return minValue + Math.floor(Math.random() * (maxValue - minValue + 1))  
}
```



Задача. Дано прибуток магазину за K тижнів. Знайти загальний прибуток

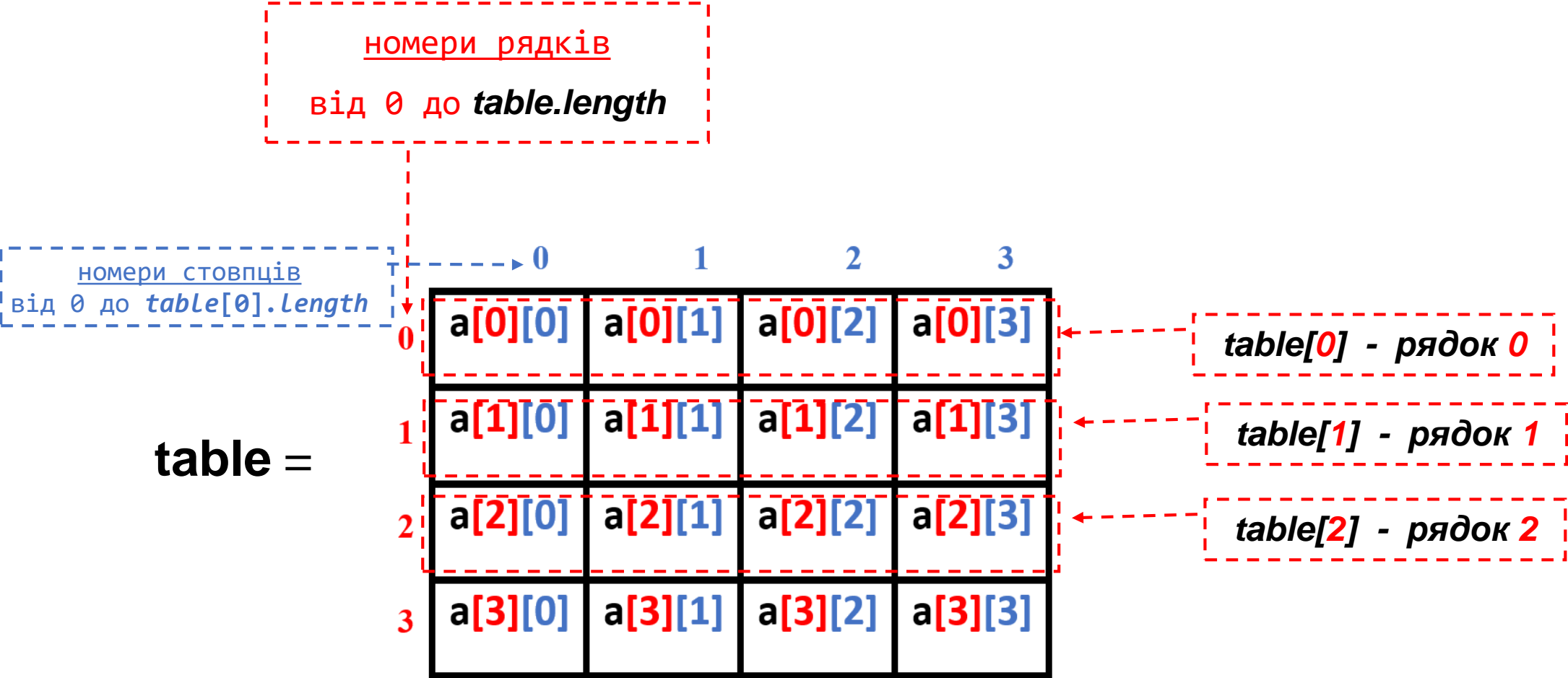
```
function getRandomNumber(minValue, maxValue) {  
  return minValue + Math.floor(Math.random() * (maxValue - minValue + 1))  
}  
  
//----- функція генерування одного одновмірного масиву (одного рядка)  
function generateRandomArray(arrayLength, minValue, maxValue) {  
  const arr = []  
  for (let i = 0; i < arrayLength; i++) {  
    const randNum = getRandomNumber(minValue, maxValue)  
    arr.push(randNum)  
  }  
  return arr  
}
```

Задача. Дано прибуток магазину за K тижнів. Знайти загальний прибуток

```
function getRandomNumber(minValue, maxValue) {  
    return minValue + Math.floor(Math.random() * (maxValue - minValue + 1))  
}  
  
//----- функція генерування одного одновмірного масиву (одного рядка)  
function generateRandomArray(arrayLength, minValue, maxValue) {  
    const arr = []  
    for (let i = 0; i < arrayLength; i++) {  
        const randNum = getRandomNumber(minValue, maxValue)  
        arr.push(randNum)  
    }  
    return arr  
}  
  
//----- функція генерування двовимірного масиву (двовимірної таблиці даних)  
function generateRandomTable(rowsNumber, columnsNumber, minValue, maxValue) {  
    const table = [] //Спочатку таблиця порожня (немає жодного рядка)  
    for (let i = 0; i < rowsNumber; i++) {  
        const randomRow = generateRandomArray(columnsNumber, minValue, maxValue) //Генеруємо рядок  
        table.push(randomRow) //Додаємо рядок у таблицю  
    }  
    return table  
}
```

Задача. Дано прибуток магазину за K тижнів. Знайти загальний прибуток  
(потрібно просумувати усі рядки (у кожному рядку – усі елементи (стовпці)

Для прямокутних масивів, де кількість елементів у рядках є однаковою, номери стовпців у кожному рядку - однакові



**for**(номер\_рядка=**поч.ряд.**; номер\_рядка <= **кінц.ряд.**; номер\_рядка += **крок**) {

**for**(номер\_стовпця = **поч.стовп.**; номер\_стовпця <= **кінц.стовп.**; номер\_стовпця += **крок**) {

.... операції з елементом : масив **номер\_рядка** **номер\_стовпця** ....

}

}

//----- Знаходження загальної суми

function getTotalSum(table) {

let sum = 0

for (let rowNumber = 0; rowNumber < table.length; rowNumber++) {

for (let colNumber = 0; colNumber < table[rowNumber].length; colNumber++) {

sum += table[rowNumber][colNumber]

}

}

return sum

}

Як змінюються  
номери рядків

Як змінюються  
номери стовпців

Що робити з  
кожним із  
елементів

Задача. Дано прибуток магазину за K тижнів. Знайти загальний прибуток

```
//----- Знаходження загальної суми -----  
function getTotalSum(table) {  
    let sum = 0  
    for (let rowNumber = 0; rowNumber < table.length; rowNumber++) {  
        for (let colNumber = 0; colNumber < table[rowNumber].length; colNumber++) {  
            sum += table[rowNumber][colNumber]  
        }  
    }  
    return sum  
}  
  
//=====  
let table = generateRandomTable(4, 3, 1, 30)  
console.log(table)  
  
let totalSum = getTotalSum(table)  
console.log(`Загальна сума = ${totalSum}`)
```

Кількість рядків (кільк. елементів таблиці)

Кількість стовпців (елементів у рядку)

Елемент, що знаходиться  
у рядку rowNumber  
і стовпці colNumber

*for*( рядок *of* таблиця ) {

Перегляд елементів з використанням *for..of*

*for*(елемент\_рядка *of* рядок ){

.....

.... операції з елементом\_рядка .....

.....

}

}

**for( рядок of таблиця ) {**

Перегляд елементів з використанням **for..of**

**for(елемент\_рядка of рядок ){**

.....  
.... операції з елементом\_рядка .....  
.....

**}**

**}**

```
//----- Знаходження загальної суми
function getTotalSum(table) {
  let sum = 0

  for (let row of table) {
    for (let element of row) {
      sum += element
    }
  }
  return sum
}
```

```
let totalSum = getTotalSum(table)
console.log(`Загальна сума = ${totalSum}`)
```

Рядки поступово беруться з таблиці

Елементи поступово вибираються з рядків

Що робити з  
кожним із  
елементів

Знаходження суми елементів двовимірного масиву (таблиці даних) з використанням *reduce*

```
let totalSum = table.reduce(  
  (prevTotalSumSum, row) => prevTotalSumSum  
    +  
    row.reduce(  
      (prevRowSum, el) => prevRowSum + el  
    ),  
  0  
)
```

Початкове значення загальної  
суми *prevTotalSum*

Знаходження суми елементів у  
рядку *row*



Задача. Дано прибуток магазину за K тижнів. Знайти загальний прибуток за робочі дні ( 0 – 4 )  
(усі рядки, у кожному рядку елементи від 0 до 4 (а бо ж < 5))

```
//----- Знаходження загальної суми за робочі дні
function getWorkingDaysSum(table) {
    let sum = 0
    for (let rowNumber = 0; rowNumber < table.length; rowNumber++) {
        for (let colNumber = 0; colNumber < 5; colNumber++) {
            sum += table[rowNumber][colNumber]
        }
    }
    return sum
}
```

**for**(номер\_рядка=**поч.ряд.**; номер\_рядка <= **кінц.ряд.**; номер\_рядка += **крок**) {

**for**(номер\_стовпця = **поч.стовп.**; номер\_стовпця <= **кінц.стовп.**; номер\_стовпця += **крок**) {

.... операції з елементом : масив **номер\_рядка** **номер\_стовпця** ....

}

//----- Знаходження загальної суми за робочі дні

function getWorkingDaysSum(table) {

let sum = 0

for (let rowNumber = 0; rowNumber < table.length; rowNumber++) {

for (let colNumber = 0; colNumber < 5; colNumber++) {

sum += table[rowNumber][colNumber]

}

}

return sum

}

Як змінюються  
номери рядків

Як змінюються  
номери стовпців

Що робити з  
кожним із  
елементів

Задача. Дано кількість відвідувачів кафе протягом року (для кожного місяця за кожен день). Визначити :

- 1) загальну кількість відвідувачів;
- 2) кількість відвідувачів за літо
- 3) кількість відвідувачів за друге півріччя

Задача. Дано зріст учнів у школі (у кожному із класів кількість учнів може бути різною). Знайти :

1) найвищого учня у школі

2) найвищого учня у кожному класі

номери рядків  
від 0 до *table.length*

Масив з різною кількістю елементів у рядках

**table =**

	0	1	2	3
0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
1	a[1][0]	a[1][1]		
2	a[2][0]	a[2][1]	a[2][2]	
3	a[3][0]			

номери стовпців (0-3)  
від 0 до *table[0].length*

номери стовпців (0-1)  
від 0 до *table[1].length*

номери стовпців (0-2)  
від 0 до *table[2].length*

номери стовпців (0-0)  
від 0 до *table[3].length*

Приклад масиву з різною довжиною рядків:

```
let table = [  
  [11, 2, 3, 6],  
  [32, 7],  
  [65, 2, 78],  
  [55]  
]
```

## Копіювання двовимірних масивів

1) Поелементно з використанням циклу *for*

```
let table = [ [11, 2, 3], [2, 89, 45] ]  
//--- Копіювання з використанням циклу for  
let newTable = [] //Спочатку новий масив порожній  
for (let rowIndex = 0; rowIndex < table.length; rowIndex++) {  
  let newRow = [] //Створюємо новий рядок  
  for (let colIndex = 0; colIndex < table[rowIndex].length; colIndex++) {  
    newRow.push(table[rowIndex][colIndex]) //Додаємо у новий рядок елементи  
  }  
  newTable.push(newRow) //Додаємо новий рядок у новий масив  
}
```

2) З використанням циклу *for..of*

```
let table = [ [11, 2, 3], [2, 89, 45] ]  
//--- Копіювання з використанням циклу for..of  
let newTable = [] //Спочатку новий масив порожній  
for (const row of table) {  
  newTable.push([...row])  
}
```

## Копіювання двовимірних масивів

2) з використанням map

```
let table = [ [11, 2, 3], [2, 89, 45] ]  
// --- (новий масив з копіями рядків)  
let newTable = table.map((row) => [...row])
```

## Копіювання двовимірних масивів

3) З використанням перетворень методів JSON

Перетворює рядок  
у масив (String)

Перетворює масив  
у рядок (String)

`новий_масив = JSON.parse( JSON.stringify ( базовий_масив ) )`

```
let table = [ [11, 2, 3], [2, 89, 45] ]  
  
let newTable = JSON.parse(JSON.stringify(table))
```

```
let table = [ [11, 2, 3], [2, 89, 45] ]
```

```
JSON.stringify(table)
```

```
'[[11,2,3],[2,89,45]]'
```

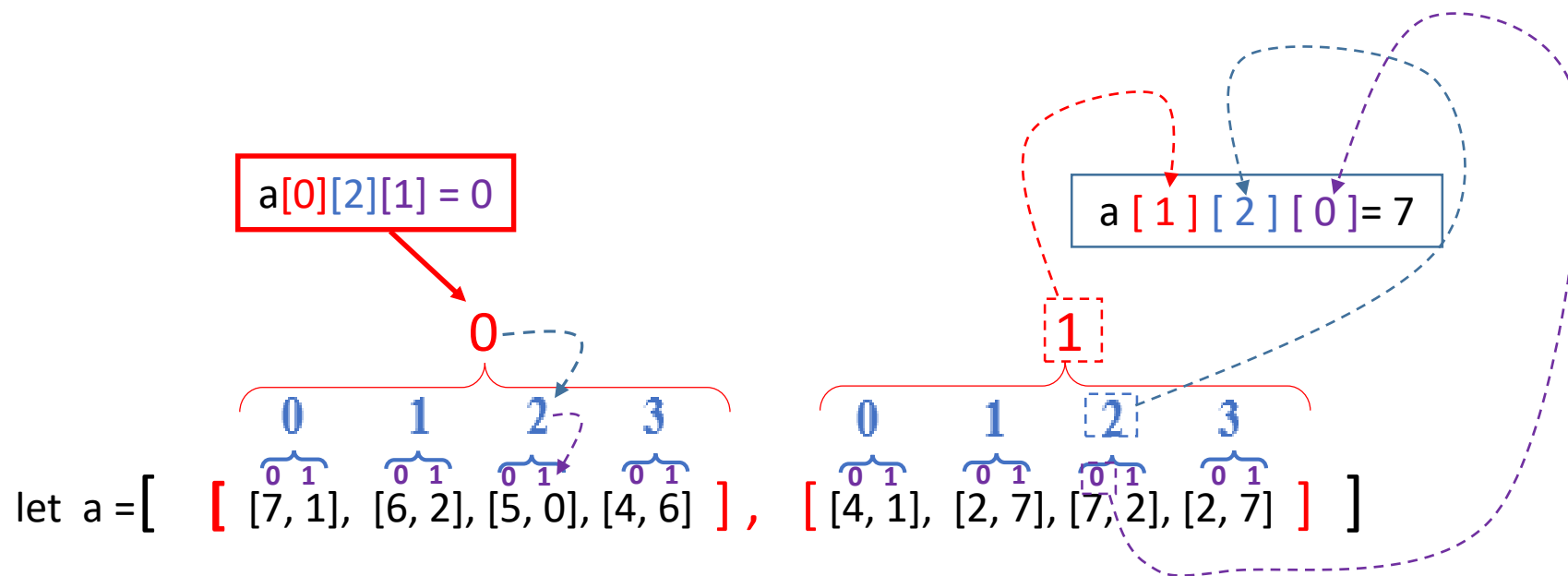
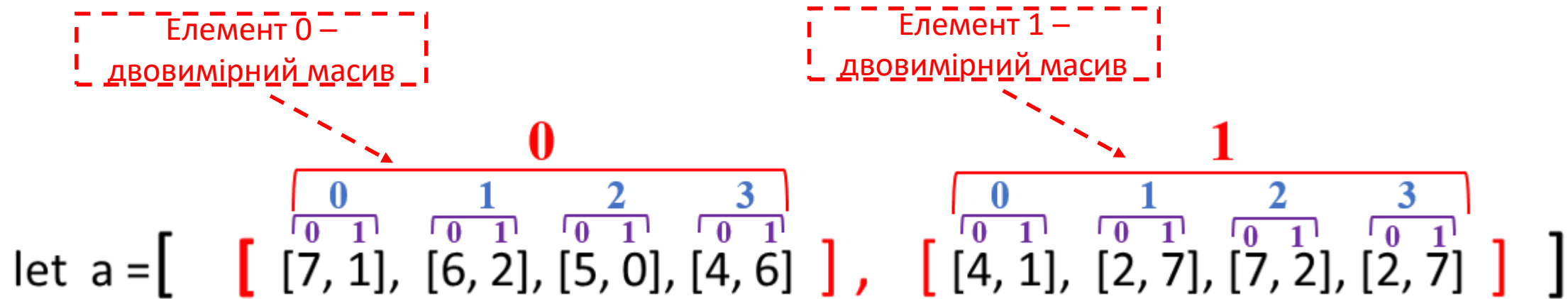
```
JSON.parse('[[11,2,3],[2,89,45]]')
```

```
[ [11, 2, 3], [2, 89, 45] ]
```



## Тривимірні масиви

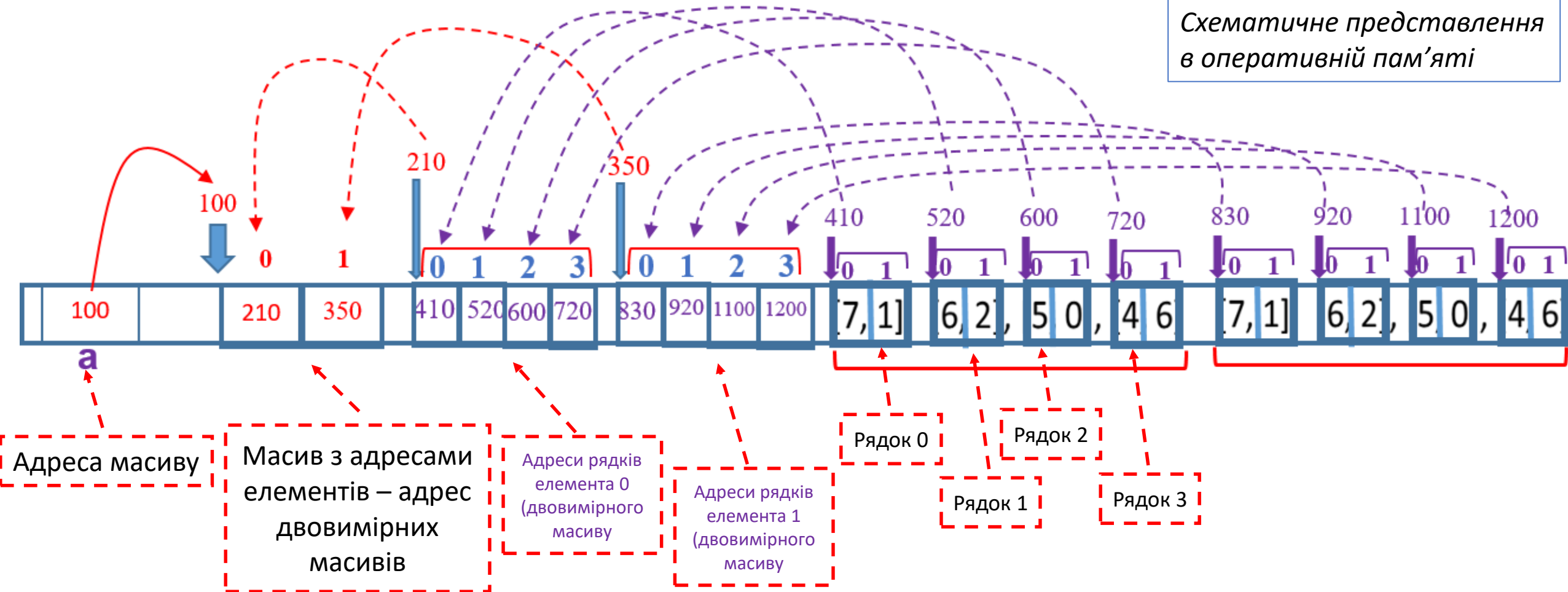
(одновимірний масив, що складається з двовимірних масивів)



# Тривимірні масиви

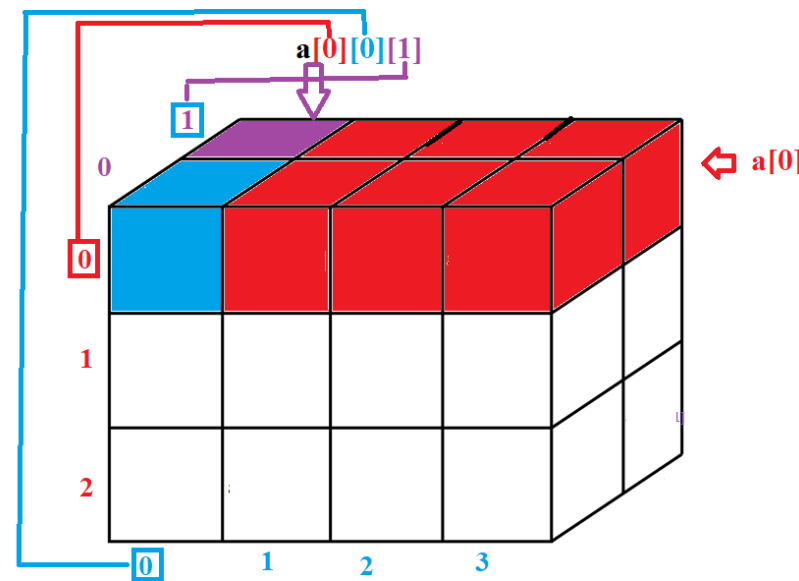
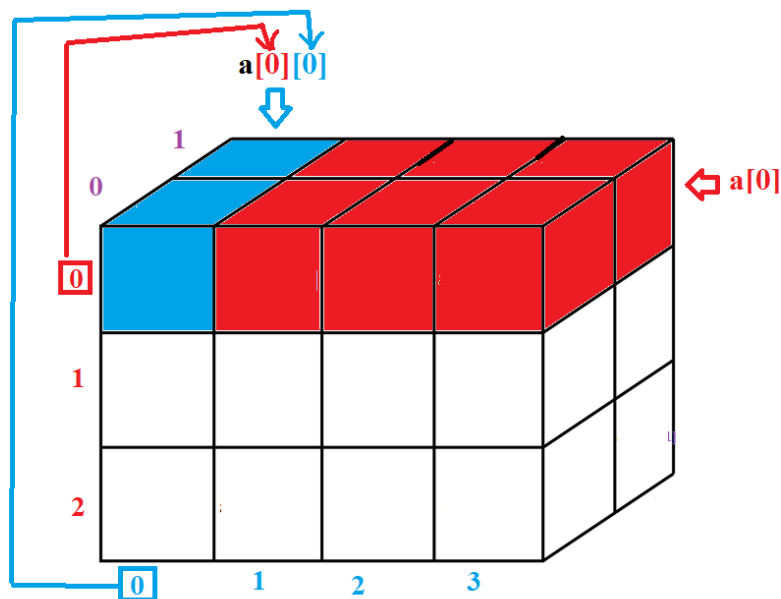
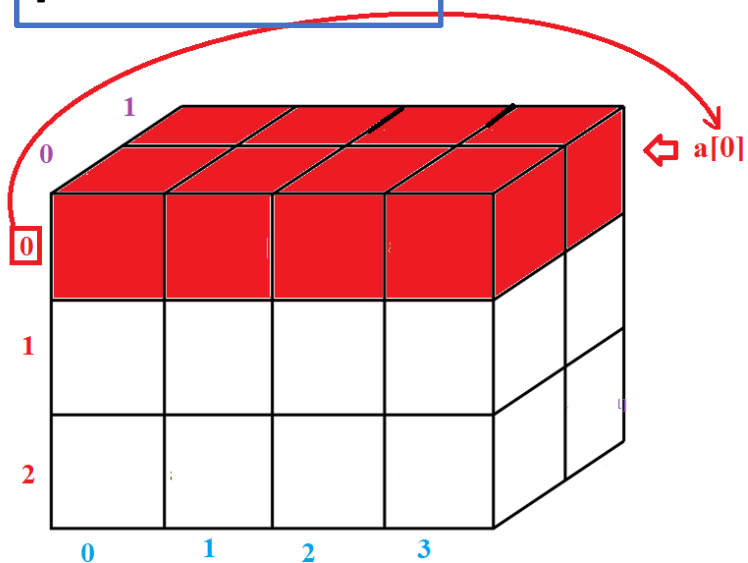
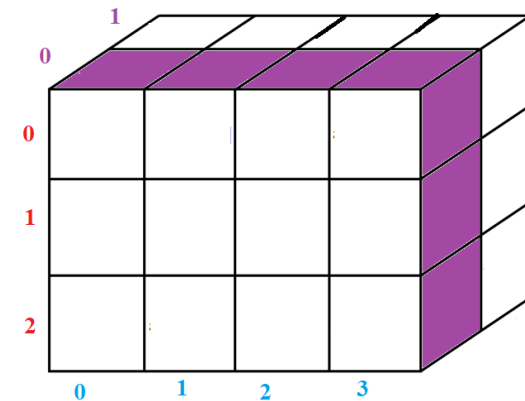
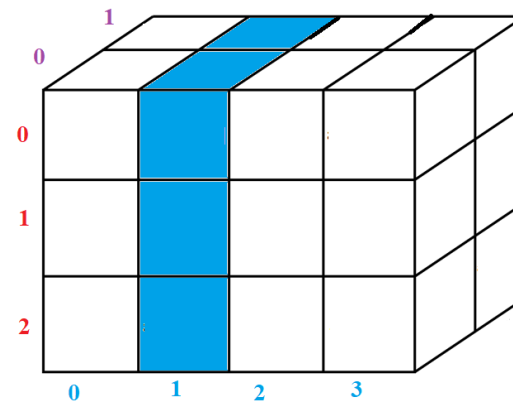
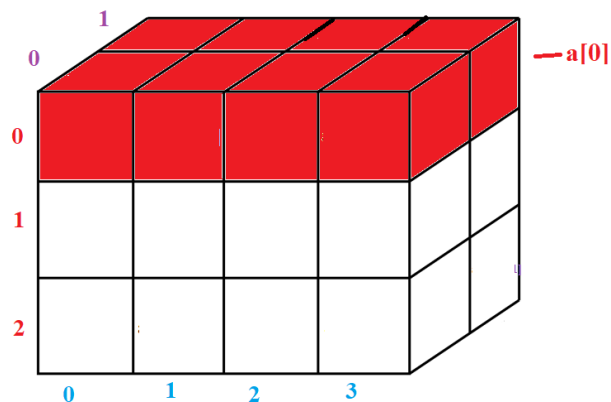
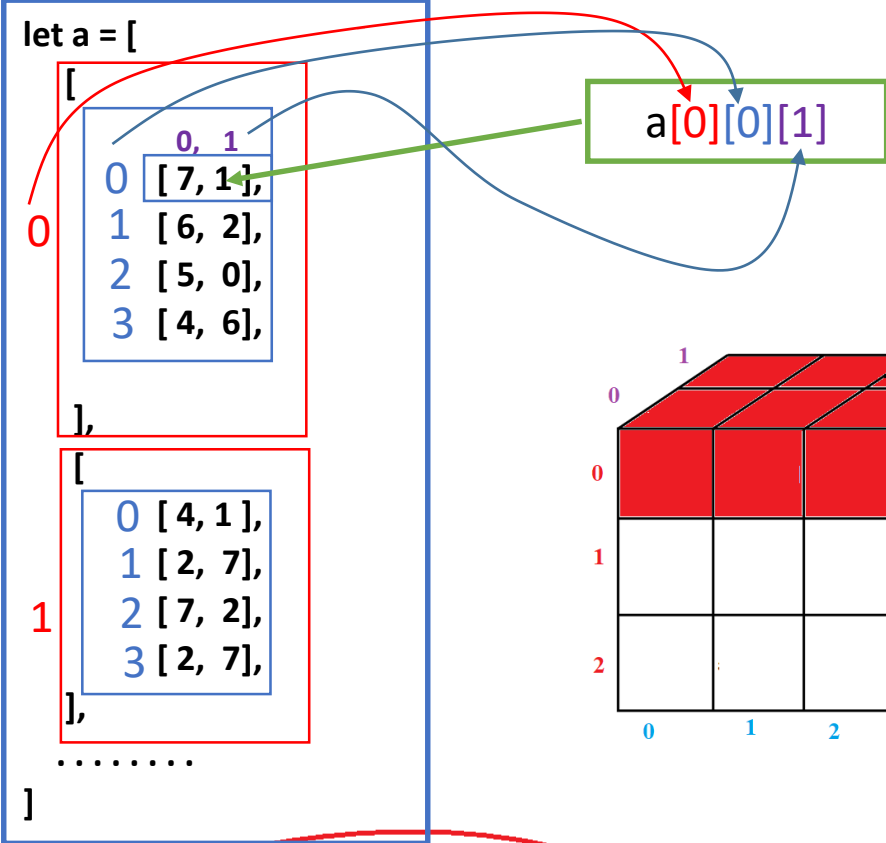
let a = [ [ [ 0 1 2 3 ] , [ 0 1 2 3 ] ] ]

Схематичне представлення  
в оперативній пам'яті

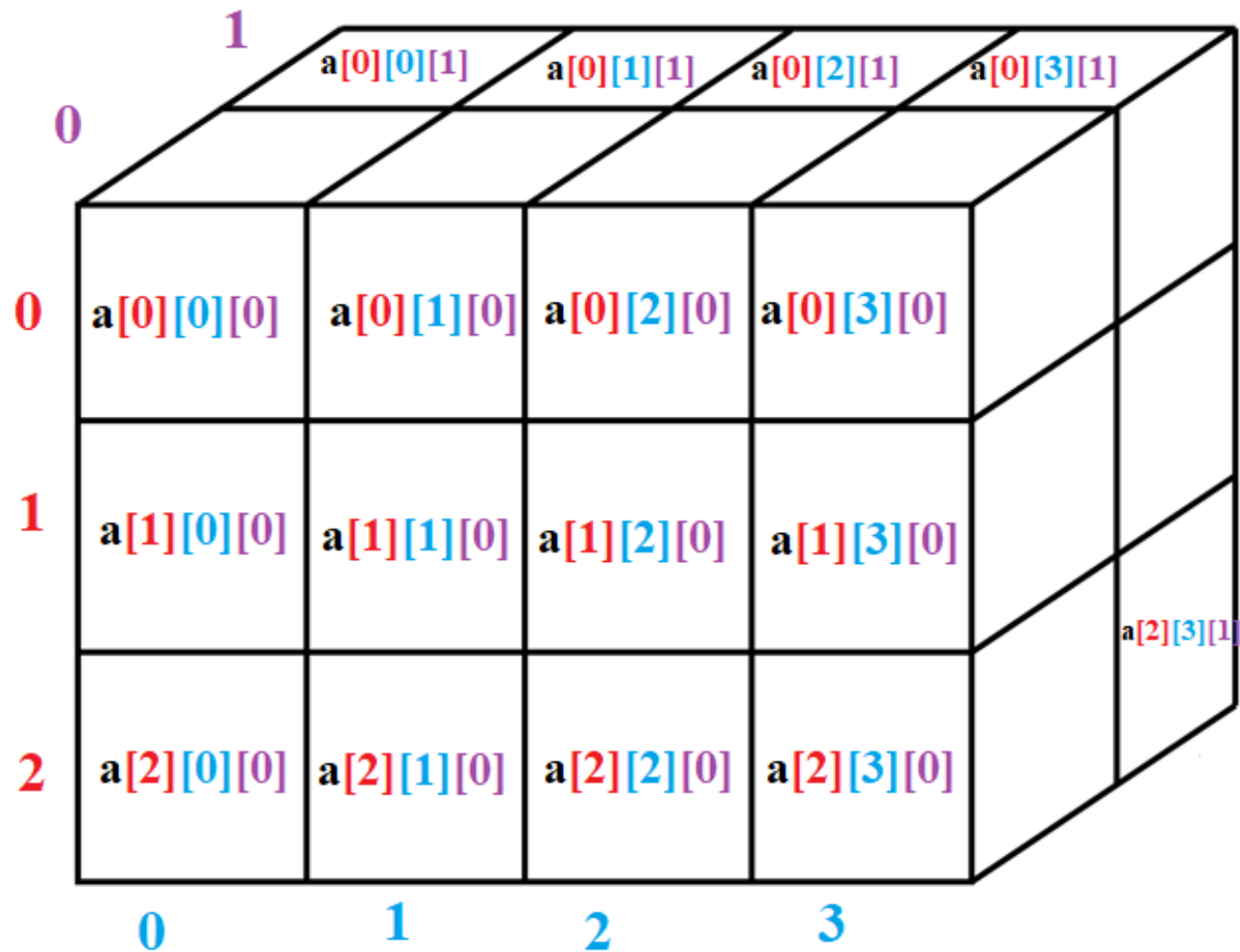
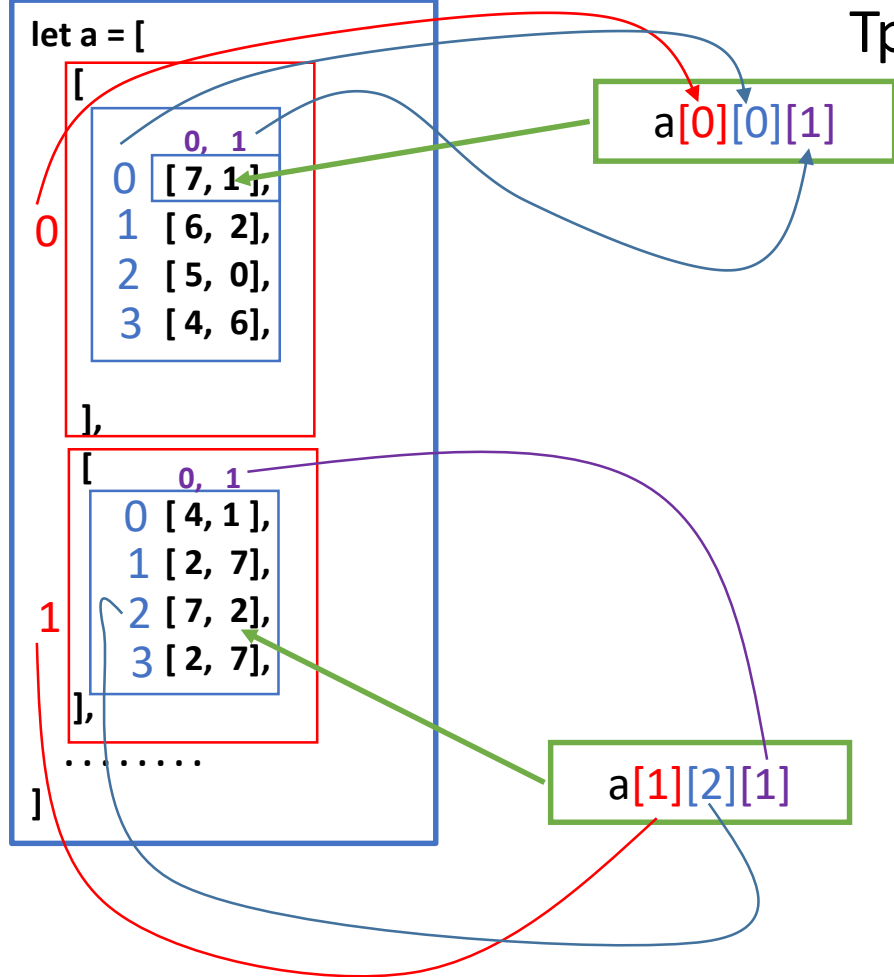


# Тривимірні масиви

(одновимірний масив , що складається з двовимірних масивів)



# Тривимірні масиви



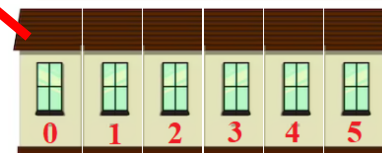
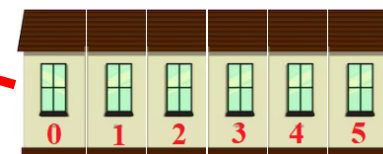
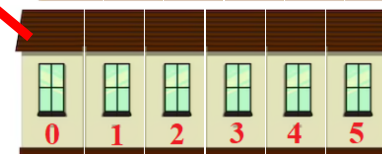
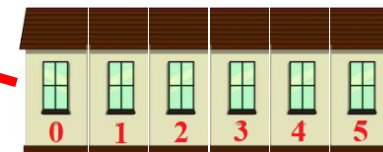
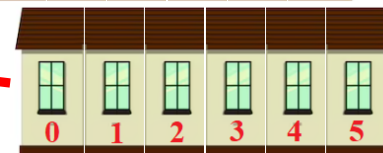
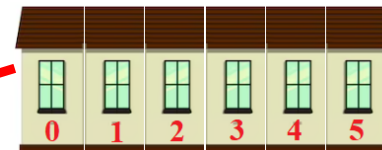
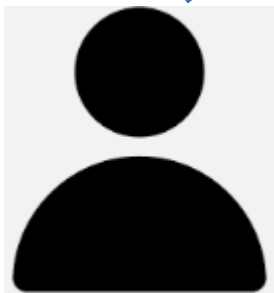
Власник знає адресу  
поштової скриньки з  
адресами груп  
готелів

Адреси готелів  
групи мережі

Адреси  
груп

У поштовій  
скриньці  
мітяться адреси  
груп готелів

У поштовій  
скриньці мітяться  
адреси готелів  
однієї групи



У місті є  $K$  шкіл по 11 класів у кожній. Відомі величини зросту учнів. Знайти найвищого учня

Дано величини прибутку від кожного працівника рекламної компанії, що складається з  $K$  філіалів.  
У кожному філіалі є певна кількість працівників (у кожному філіалі кількість працівників може бути різною).  
Знайти загальний прибуток та прибуток у кожному із філіалів.

## Зменшення вимірності багатовимірних масивів

Призначення	Загальна форма	Приклади
Метод <b>flat</b> призначений для створення нового масиву з меншою вимірністю елементів <b>depth</b> (за замовчуванням зменшення вимірності на 1 одиницю <b>depth=1</b> )	<code>назва_масиву.flat(depth)</code>	<pre>const arr1 = [0, 1, 2, [ 3, 4 ]] console.log(arr1.flat()) //видал.1 дужка [0, 1, 2, <del>[ 3, 4 ]</del>] // Expected output: [0, 1, 2, 3, 4]  //----- const arr2 = [0, 1, 2, [ [ [ 3, 4 ] ] ] ]; console.log(arr2.flat(2)) //видал.2 дужки [0, 1, 2, <del>[</del> <del>[</del> <del>[ 3, 4 ]</del> <del>]</del> <del>]</del> <del>]</del> ] // Expected output: [0, 1, 2, [3, 4]]  //----- Якщо хочемо отримати одновимірний масив незалежно від вкладеності елементів масиву, то можна використати <b>Infinity</b>  const arr4 = [1,2, [3,4, [5,6, [7,8, [9,10]]]]]; arr4.flat(Infinity);  // [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]</pre>



Дано покази температур за K тижнів. Сформувати список, що містить тільки додатні покази.

Дано зріст учнів у K школах. Сформувати список, тільки з тих величин зросту, які більші за 150см

Дано покази продажу автомобілів компанії за країнами, областями, містами і моделями.

(поки зберігаємо за номерами (номер країни, номер області, номер міста, номер моделі) як 4-вимірний масив)

Знайти:

- 1) загальну кількість проданих авто
- 2) загальну кількість проданих моделей кожного виду
- 3) загальну кількість проданих авто у кожній із країн
- 4) загальну кількість проданих моделей у кожному із міст