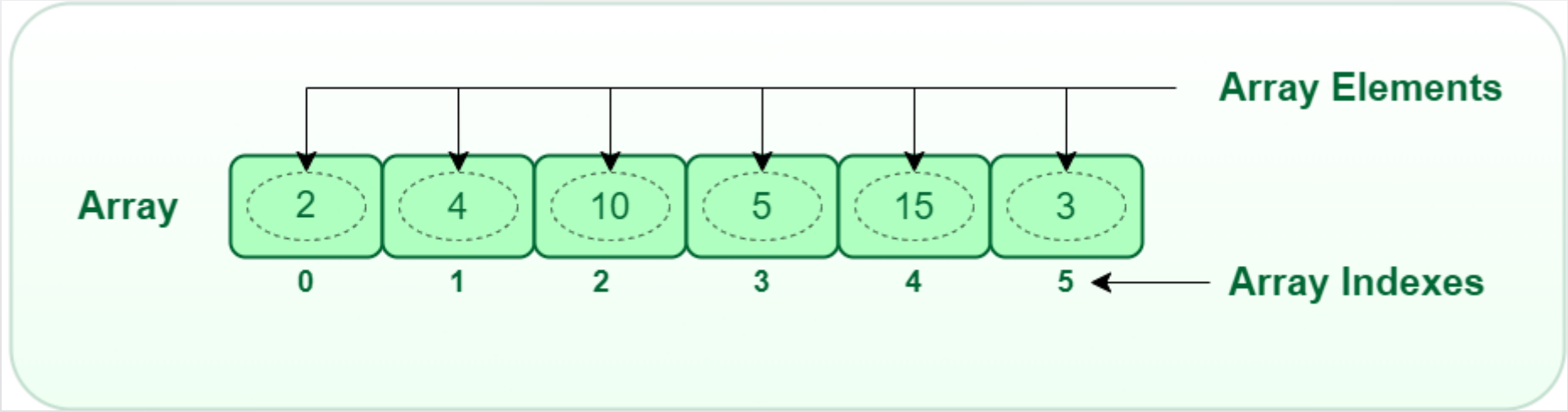
DATA STRUCTURES

Lab 2: Array 1

Lecturer: Hussien Omer AL_Baiti

What is array

An array is a collection of similar data elements stored at contiguous memory locations.



Build an Array Algorithm 1

Step 1 : start

Step 2 : read length of array (size)

Step 3 : define array (int [ ] items = new int [size])

Step 4 : init i = 0

Step 5 : loop until i == size

Step 5.1 : items [i] = read new item

Step 5.2 : i++

Step 6 : end

Build an Array Algorithm 2

Step 1 : start

Step 2 : read size of array (size)

Step 3 : build array called (ptr)

Step 4 : init i = 0

Input

Step 5 : while i < size

Step 5.1 : ptr [i] = read new item

Step 5.2 : i++

Process

Step 5.3 : print ptr array → **Output**

Step 6 : end

Some (Python & C#) functions

Task	Python Code	C# Code	Description C# Code
Print message or object	Print ( )	Console.Write ( )	Show value in one line
		Console.WriteLine ( )	Show value in one line, and go to new line
Enter values by users	Input ( )	Console.ReadLine()	Allow to the user add value by console window to variables
Casting	Int(input())	Convert.ToInt32(Console.ReadLine())	Convert numbers that users enterd by console window to integer type
Stop console window	-	Console.ReadKey();	-

C# code to Build an Array

```
Console.WriteLine("enter size of array");

int size = Convert.ToInt32(Console.ReadLine());

int[] ptr = new int[size];

for ( int i = 0 ; i< size ; i++ )
{
    ptr[i]= Convert.ToInt32(Console.ReadLine());
}

for (int i = 0; i < ptr.Length; i++)
{
    Console.WriteLine(ptr[i]);
}
Console.ReadKey();
```

Create an Array

- Arrays are used to store multiple values in a single variable

```
string[] cars;
```

```
int[] myNum = {10, 20, 30, 40};
```

```
string[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
```

Access the Elements of an Array

- You access an array element by referring to the index number.

```
string[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
Console.WriteLine(cars[0]);
// Outputs Volvo
```

Change an Array Element

- To change the value of a specific element, refer to the index number:

```
string[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
Console.WriteLine(cars[0]);
// Outputs Volvo
```

```
cars[0] = "Opel";
```

Array Length

- To find out how many elements an array has, use the Length property:

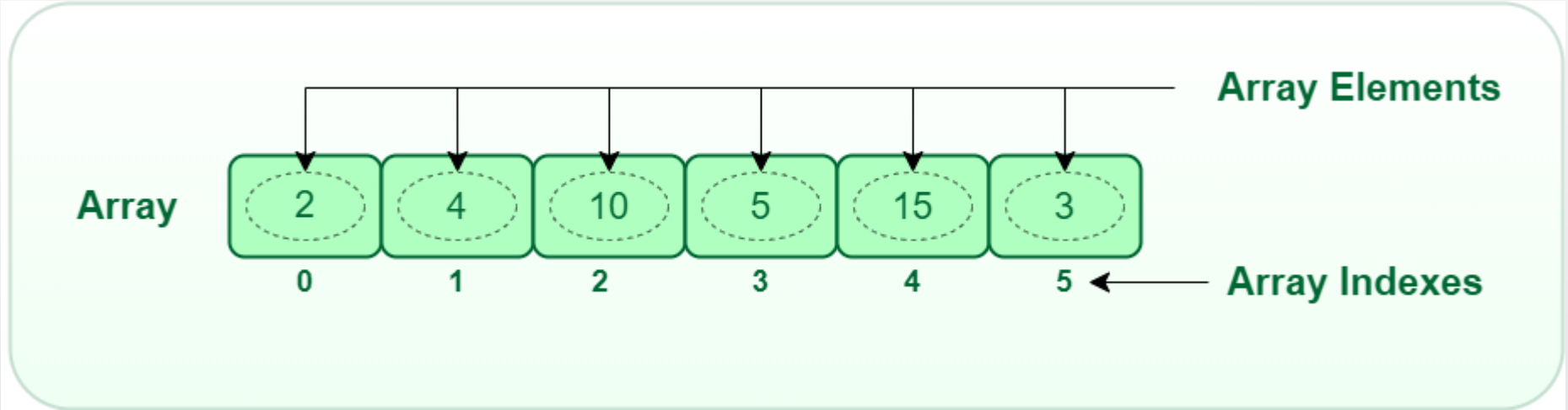
```
string[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
Console.WriteLine(cars.Length);
// Outputs 4
```

Other Ways to Create an Array

```
// Create an array of four elements, and add values later
string[] cars = new string[4];

// Create an array of four elements and add values right away
string[] cars = new string[4] {"Volvo", "BMW", "Ford", "Mazda"};

// Create an array of four elements without specifying the size
string[] cars = new string[] {"Volvo", "BMW", "Ford", "Mazda"};

// Create an array of four elements, omitting the new keyword, and without specifying the size
string[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
```

Basic Array Operations

- Traverse
- Insertion
- Deletion
- Search
- Update



Traversing

- Just like lists, we can access elements of an array by **indexing**, and **looping**.

1) Indexing

```
string[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
Console.WriteLine(cars[0]);
// Outputs Volvo
```

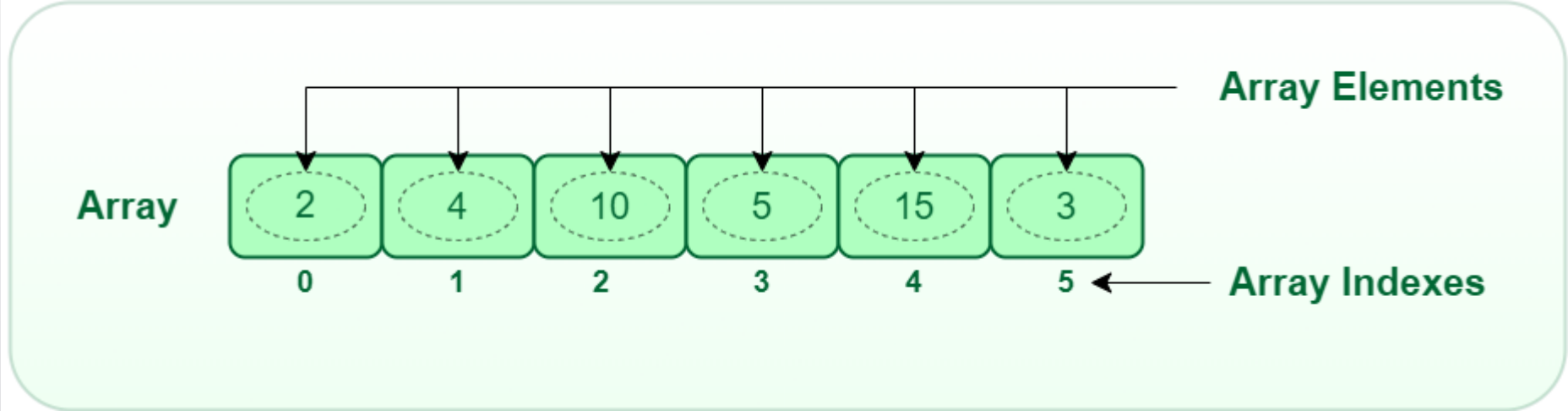
```
cars[0] = "Opel";
```

2) Looping

```
string[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
for (int i = 0; i < cars.Length; i++)
{
  Console.WriteLine(cars[i]);
}
```

Search & Update

- When we execute this operations we need to search by **value** or **position** , and you can change value or show it.



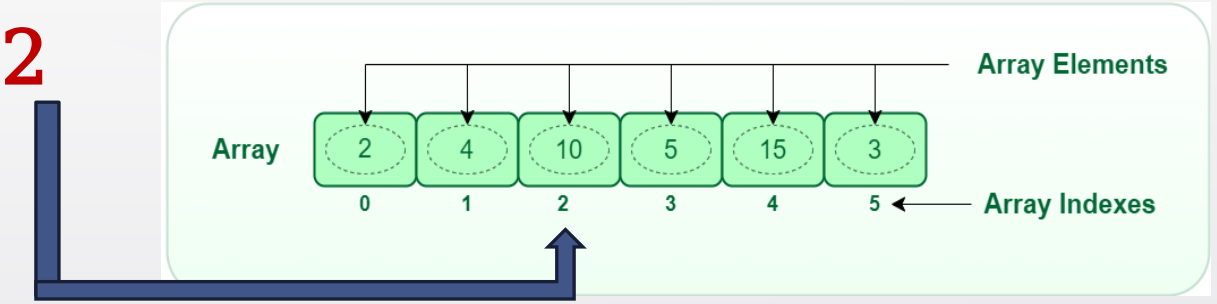
Home work 1

Write **algorithm** and **C# code** of search integer item in array:

- user read array size.
- user read array items.
- user read item that search about it.
- If exist in array print message called **is found** and print it’s index .
- Else print message called **is not found .**

Cont..

Example : if I search about **10** number, when read this number, you will show **is found** in **index 2**



if I search about **100** number, when read this number, you will show **is not found** .

THE END