

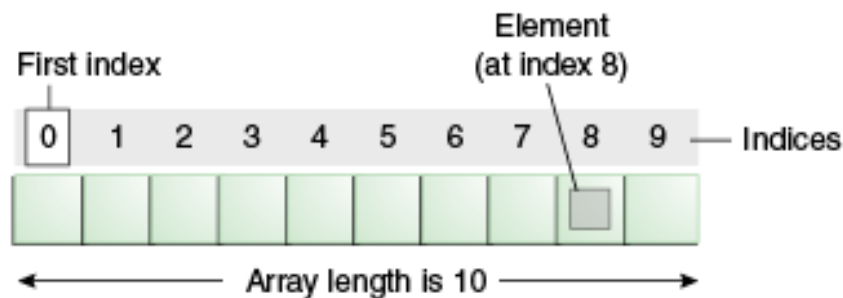


1

Mảng 1 chiều C#

2

Ví dụ vận dụng



1

Mảng 1 chiều C#

❑ **1. Khái niệm :** *Mảng là tập hợp các đối tượng có cùng kiểu dữ liệu.
_Chúng có kích thước cố định không thể thay đổi*

❑ **2 . Tại sao phải dùng mảng:**

Ví dụ : Chúng ta có khoảng 50 điểm của học sinh cần lưu, nếu không dùng mảng thì chúng ta phải khai báo 50 biến float

=> Gom nhóm các đối tượng có chung tính chất lại với nhau giúp code gọn gàng hơn.

❑ **3. Khai báo mảng :**

kiểu_dữ_liệu[] <tên mảng>;

Ex : *int[] stt;*

string[] ten;

❑ **4. Khởi tạo mảng:** **<tên mảng> = new kiểu_dữ_liệu[số_phần_tử]**

Ex 2 : *stt = new int[10];* // mảng sẽ 10 phần tử là các số 0

Ex 3: *int[] Mang = new int[4];*



1

Mảng 1 chiều C#

❑ 5 . Khởi tạo và gán giá trị cho mảng :

Kiểu dữ liệu[] tên_mảng = new Kieudulieu[Spt]{pt1,pt2,pt3...};

int[] mang2 = new int[2] { 1, 2 };

Rút ngắn : *int[] mangVdidu = {1,2,3,4,5,6,7,8,9};*

❑ 6. Khởi tạo mảng ngẫu nhiên :

```
Random r = new Random();
int[] mang3 = new int[3] { r.Next(11), r.Next(101), r.Next(51) };
```

❑ 7. Length : trả về số phần tử của mảng

```
int doDai = mang2.Length;
Console.WriteLine("Số phần tử của mảng mang2 là: " + doDai);
```

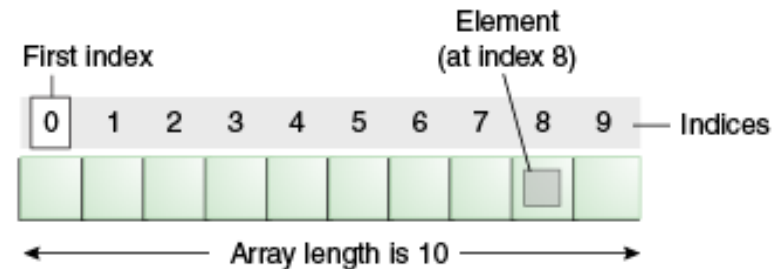


1

Mảng 1 chiều C#

❑ 8 . Truy xuất phần tử qua index :

```
int[] mang4 = new int[3] { 1, 3, 5 };
Console.WriteLine(mang4[0]);
Console.WriteLine(mang4[1]);
Console.WriteLine(mang4[2]);
Console.WriteLine(mang4[mang4.Length - 1]);
// xuất ký tự cuối cùng dùng length
```



❑ 9. Thay đổi giá trị phần tử của mảng bằng cách gọi index :

```
int[] mang5 = new int[3] { 1, 3, 5 };
Console.WriteLine("Giá trị trước khi đổi = " + mang5[1]);
mang5[1] = 9;
Console.WriteLine("Giá trị sau khi đổi = " + mang5[1]);
```



1

Mảng 1 chiều C#

❑ 10 . duyệt mảng for (duyệt theo index):

```
int[] mang6 = new int[3] { 1, 3, 5 };
for (int i = 0; i < mang6.Length; i++)
{
    //Console.WriteLine(i);
    Console.WriteLine(mang6[i]);
}
```

❑ 11. duyệt mảng dùng foreach (duyệt theo phần tử) :

```
int[] mang7 = new int[3] { 1, 3, 5 };
foreach (int i in mang7)
{
    Console.WriteLine(i);
}
```

❑ 12. Phép gán mảng :

**** gán mảng mới = mảng đã tồn tại (tham chiếu cùng vùng nhớ)**

**chú ý : khi thay đổi giá trị của 1 mảng, thì mảng còn lại cũng thay đổi ,
// do mảng là kiểu tham chiếu, nó sẽ tham chiếu đến 1 vùng nhớ.**

```
int[] mang8 = new int[3] { 1, 3, 5 };
int[] mang9 = mang8; // khai báo 1 mảng mới = mang8
Console.WriteLine("giá trị trước khi đổi: " + mang8[0]);
Console.WriteLine("giá trị trước khi đổi: " + mang9[0]);
mang9[0] = 100;
Console.WriteLine(mang9[0]);
Console.WriteLine(mang8[0]);
```



1

Mảng 1 chiều C#

❑13 . CopyTo (copy ra mảng mới trên vùng nhớ mới)

Array_nguồn.CopyTo(Array_đích, int_index);

//copy mảng nguồn vào mảng đích đến 11, copy đè lên index 2 của mảng đích

```
//10. CopyTo (copy đè giá trị vào mảng đã có)
int[] mang10 = { 100, 300, 500, 700, 900 };
int[] mang11 = {1,2,3,4,5,6,7,8,9};
// khởi tạo mảng mới có số phần tử = mảng muốn copy
mang10.CopyTo(mang11, 2); // copy 10 đến 11, copy đè lên index 2 của mảng đích
Console.WriteLine("Mảng 11 sau copyTo là: ");
```

14. Copy(Array_nguồn, Array_đích, Int32) (copy ra mảng mới trên vùng nhớ mới)

```
int[] mang12 = { 1, 3, 5, 9, 8, 7 };
int[] mang13 = new int[5];
Array.Copy(mang12, mang13, 4); // copy 4 phần tử
Console.WriteLine("Mảng 13 sau copy là: ");
```

❑15. Clone (tạo ra mảng mới trên vùng nhớ mới)

```
int[] mang14 = { 1, 3, 5, 9, 8, 7, 2 };
int[] mang15 = (int[])mang14.Clone();
```



1

Mảng 1 chiều C#

❑ **16 . Array.Reverse()** : Đảo ngược mảng

```
int[] mang16 = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };
Array.Reverse(mang16);
// mảng sẽ bị đổi thành mang16 = {9,8,7,6,5,4,3,2,1}
```

17. Array.Sort() (Sắp xếp mảng tăng dần)

```
int[] mang17 = { 1, 9, 4 };
Array.Sort(mang17);
//trả về : mang17={1,4,9}
```



2

Ví dụ vận dụng

- ✓ 1. Viết chương trình tạo 1 mảng 1 chiều gồm các phần tử là số nguyên, có n phần tử ngẫu nhiên, n do người dùng nhập từ bàn phím
- ✓ 2. Xuất các giá trị trong mảng
- ✓ 3. Đảo ngược mảng, và xuất mảng sau khi đảo ngược
- ✓ 4. Sắp xếp mảng tăng dần
- ✓ 5. Tính tổng các phần tử trong mảng
- ✓ 6. Cho người dùng nhập 1 số bất kỳ, kiểm tra số đó có tồn tại trong mảng hay không, nếu có xuất ra vị trí index của số đó trong mảng

