

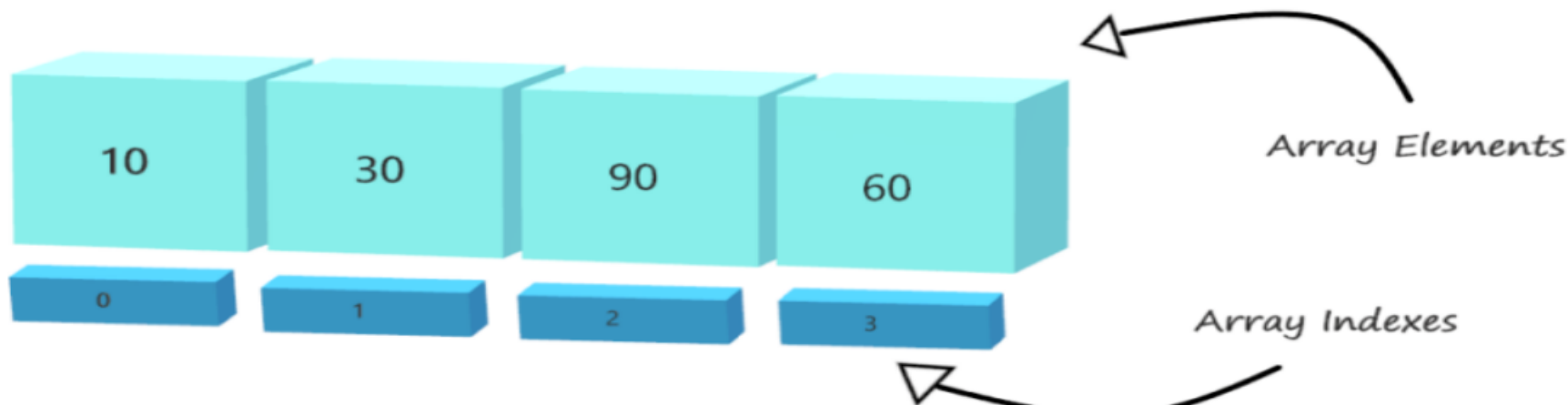


Mảng 1 chiều(1D ARRAY)

Mảng 1 Chiều Trong C++

Mảng 1 chiều là gì?

1. Mảng 1 chiều là một cấu trúc dữ liệu gồm nhiều phần tử có cùng kiểu dữ liệu, được lưu trữ ở các ô nhớ liên tiếp nhau trong bộ nhớ.
2. Mảng 1 chiều được sử dụng khi bạn cần lưu trữ một số lượng lớn các phần tử có cùng kiểu dữ liệu. Ví dụ: 100 số nguyên, 1000 số thực, 1000 học sinh,...
3. Mảng 1 chiều đơn giản, dễ hiểu và được sử dụng rất nhiều trong mọi ngôn ngữ lập trình.





1

Cách Khai Báo Mạng

Cú Pháp Và Ví Dụ

Cách khởi tạo 1 mảng đơn giản trong C++:

```
Data_type array_name [Number_of_element]
```





2

Cách Truy cập phần tử và duy ệт mảng

Các phần tử trong mảng được truy cập thông qua chỉ số, chỉ số của mảng được đánh từ 0 và kết thúc bởi $n - 1$ với n là số lượng phần tử của mảng.

Cú Pháp:

```
array_name[index];
```

Ví dụ :

```
int[6] a = {5, 2, 4, 1, 7, 3};
```

```
a[0] = 5, a[1] = 2, a[2] = 4, a[3] = 1, a[4] = 7, a[5] = 3
```


Cách truy cập phần tử và duyệt mảng

Duyệt bằng chỉ số và for-each :

```
#include <bits/stdc++.h>
using namespace std;
int main(){
    int n; // số lượng phần tử trong mảng
    cin >> n;
    int a[1000];
    // khai báo mảng a ít nhất n phần tử
    for(int i = 0; i < n; i++){
        cin >> a[i];
    }
    for(int i = 0; i < n; i++){
        cout << a[i] << ' ';
    }
}
```

```
int main(){
    int a[5] = {1, 2, 3, 4, 5};
    for(int x : a){
        cout << x << ' ';
    }
}
```

3

Những lưu ý khi sử dụng hàm.



Mảng làm tham số của hàm

- Khi mảng làm tham số của hàm, những thay đổi trong hàm sẽ làm thay đổi tới mảng mảng được truyền vào.
- Khi xây dựng hàm có tham số là mảng, cần phải kèm theo 1 tham số nữa là số lượng phần tử trong mảng.

```
#include <bits/stdc++.h>
using namespace std;
void nhanDoi(int a[], int n){
    for(int i = 0; i < n; i++){
        a[i] *= 2;
    }
}
int main(){
    int a[4] = {1, 2, 3, 4};
    nhanDoi(a, 4);
    for(int i = 0; i < 4; i++){
        cout << a[i] << ' ';
    }
}
```



4

**Các bài toán cơ bản trên
mạng.**

Liệt kê hoặc đếm các phần tử trong mảng thỏa yêu cầu.

VD: Chương trình liệt kê số nguyên tố:

```
#include <bits/stdc++.h>
using namespace std;
bool nt(int n){
    for(int i = 2; i <= sqrt(n); i++){
        if (n % i == 0) return false;
    }
    return n > 1;
}

int main(){
    int n; cin >> n;
    int a[n];
    for(int i = 0; i < n; i++){
        cin >> a[i];
    }
    for(int i = 0; i < n; i++){
        if (nt(a[i])){
            cout << a[i] << ' ';
        }
    }
}
```



Sắp xếp tăng dần.

```
#include <bits/stdc++.h>
#include <algorithm> // For sort
using namespace std;
int main(){
    int n; cin >> n;
    int a[n];
    for(int i = 0; i < n; i++) cin >> a[i];
    sort(a, a + n); // sắp xếp theo thứ tự tăng dần
    for(int i = 0; i < n; i++){
        cout << a[i] << ' ';
    }
}
```

Sắp xếp giảm dần

```
#include <bits/stdc++.h>
#include <algorithm> // For sort
using namespace std;
int main(){
    int n; cin >> n;
    int a[n];
    for(int i = 0; i < n; i++) cin >> a[i];
    sort(a, a + n, greater<int>()); // sắp xếp theo thứ tự Giảm Dần
    for(int i = 0; i < n; i++){
        cout << a[i] << ' ';
    }
}
```


Tìm kiếm sự xuất hiện của 1 phần tử trong mảng

```
#include <bits/stdc++.h>
#include <algorithm> // For sort
using namespace std;
bool check(int a[], int n, int x){
    for(int i = 0; i < n; i++){
        if(a[i] == x) return true; // found
    }
    return false; // not found
}
int main(){
    int n, x; cin >> n >> x;
    int a[n];
    for(int i = 0; i < n; i++) cin >> a[i];
    if(check(a, n, x)){
        cout << "YES" << endl;
    }
    else cout << "NO" << endl;
}
```




THANKS

Full House Tất Cả Là Một Nhà!

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**.

Please keep this slide for attribution.