**Vector**

1. Công dụng:

* Là 1 thư viện chuyên hỗ trợ việc làm việc với mảng động.
* Rất tiện ích.
* Khỏi cần dùng từ khóa new, delete để xử lý mảng, vector đã xử lý cho ta trường hợp bị leaks.

1. Syntax:

Include<vector>;

Khai báo:

Vector<datatypes> array;

* Không cần khai báo số lượng phần tử.
* Truy cập mảng trong vector:

Array[i] or array.at(i)

Vector<type> array(n, k); n là số ele, k là default.

Vector<type> array[n]; với n vector array.

1. Một số thao tác:

* Size(): Xem số lượng phần tử đang được sử dụng trong mảng.

Arr.size();

* Resize(): Thay đổi kích thước mảng.

Arr.resize(k); với k là kích thước cần thay đổi.

Khi thay đổi, các phần tử đã khai báo trước được giữ nguyên, các phần tử thêm vào sẽ là giá trị mặc định của datatype.

Nếu thay đổi ít hơn, nó sẽ loại bỏ các giá trị ở cuối.

* Capacity(): Số lượng phần tử được cấp phát cho bộ nhớ(tổng mảng).

Arr.capacity();

* Reserve(): Cấp phát dung lượng ban đầu.

Arr.reserve(k); k là dung lượng muốn cấp phát.

* V.clear(): xóa hết các pt trong vector.
* Iterater:

Vector<type>::iterater it= v.begin() or v.end(). Begin là ptu đầu tiên, end là ptu sau phần tử cuối.

1. Hành vi ngăn xếp (stack):

Vector<int> stack;

* Push\_back(): Thêm 1 phần tử ở cuối vector:

Stack.push\_back(n); với n là phần tử cần thêm.

* Back() : Trả về phần tử cuối trong vector != front()

Stack.back();

* Pop\_back(): Xóa phần tử cuối ở vector

Stack.pop\_back();