



# I. LỚP VECTOR

## ◆ BÀI TOÁN MINH HỌA

Nhập một danh sách số nguyên dương A với số phần tử không biết trước. Thao tác nhập kết thúc khi phần tử nhập vào có giá trị  $A_i \leq 0$ . In ra màn hình danh sách A, vị trí i của phần tử có giá trị k (k được nhập từ bàn phím) ở trong A và 5 giá trị lớn nhất của A. Nếu không tìm thấy k thì đặt  $i = -1$ .

Ví dụ:  $A = \{1, 2, 8, 3, 7, 4, 6, 10, 9, 21\}$ ,  $k = 4$ .

Kết quả

1 2 8 3 7 4 6 10 9 21

5        21, 10, 9, 8, 7



# I. LỚP VECTOR

## ◆ BÀI TOÁN MINH HỌA

Bài toán có thể được giải quyết bằng cách sử dụng thư viện `<vector>` và `<algorithm>` như sau:

```
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;
void NhapDS(vector<int> &);
void InDS(vector<int>);
void Top5(vector<int>);
int TimK(vector<int>, int);
```



# I. LỚP VECTOR

## ◆ BÀI TOÁN MINH HỌA

```
int main() {  
    vector<int> A;  
    int k;  
    NhapDS(A);  
    cout << "Danh sach da nhap: ";  
    InDS(A);  
    cout << "Gia tri can tim k = ";  
    cin >> k;  
    cout << TimK(A, k) << "\\t";  
    Top5(A);  
    return 0;  
}
```



# I. LỚP VECTOR

## ◆ BÀI TOÁN MINH HỌA

```
void NhapDS(vector<int> &v) {  
    int tmp;  
    cout << "Nhap danh sach" << endl;  
    cin >> tmp;  
    while (tmp > 0) {  
        v.push_back(tmp);  
        cin >> tmp;  
    }  
}
```





# I. LỚP VECTOR

## ◆ BÀI TOÁN MINH HỌA

```
void InDS(vector<int> v) {  
    for (int i = 0; i < v.size(); i++)  
        cout << v[i] << ' ';  
    cout << endl;  
}
```



# I. LỚP VECTOR

## ◆ BÀI TOÁN MINH HỌA

```
void Top5(vector<int> v) {  
    sort(v.begin(), v.end());  
    if (v.size() < 5) {  
        cout << "DS khong co du 5 phan tu" << endl;  
        return;  
    }  
    for (vector<int>::iterator i = v.end() - 1;  
        i > v.end() - 6; i--)  
        cout << *i << ' '  
    }
```



# I. LỚP VECTOR

## ◆ BÀI TOÁN MINH HỌA

```
int TimK(vector<int> v, int k) {  
    vector<int>::iterator i;  
    i = find(v.begin(), v.end(), k);  
    if (i != v.end())  
        return i - v.begin();  
    return -1;  
}
```



# I. LỚP VECTOR

## ◆ ĐỊNH NGHĨA LỚP VECTOR

vector, được định nghĩa trong <vector>, là một lớp quản lý danh sách các đối tượng cùng kiểu. Biến kiểu vector được khai báo như sau:

```
vector<kiểu> tên_biến;
```

Các biến kiểu vector có đặc điểm:

- Tương tự như mảng, truy xuất với phép toán []
- Có thể tăng kích thước khi thêm phần tử
- Có thể duyệt tuần tự theo các biến thuộc kiểu

```
vector<kiểu>::iterator
```





# I. LỚP VECTOR

## ◆ ĐỊNH NGHĨA LỚP VECTOR

Một số phương thức của lớp vector:

- **begin()** Trả về biến iterator trỏ đến phần tử đầu của vector
- **end()** Trả về biến iterator trỏ đến vị trí sau phần tử cuối của vector
- **size()** Trả về số phần tử của vector
- **push\_back()** Thêm một phần tử vào cuối vector

***Yêu cầu sinh viên tìm hiểu thêm các phương thức khác trong slide tham khảo thư viện STL.***



# I. LỚP VECTOR

## ◆ TÌM KIẾM VỚI ĐỐI TƯỢNG VECTOR

Các phần tử trong một biến vector có thể được tìm kiếm nhờ hàm `find()` (được định nghĩa trong `<algorithm>`). Hàm `find()` được sử dụng như sau:

```
vector<kiểu> A;  
vector<kiểu>::iterator i;  
int k;  
.  
.  
.  
i = find(A.begin(), A.end(), k);
```

Kết quả của hàm `find()` là một biến kiểu iterator trở tới phần tử cần tìm của vector hoặc trả về `end()` nếu không tìm thấy.



# I. LỚP VECTOR

## ◆ SẮP XẾP VỚI ĐỐI TƯỢNG VECTOR

Có thể sắp xếp các phần tử trong một biến kiểu vector bằng hàm `sort()` (được định nghĩa trong `<algorithm>`). Hàm `sort()` được sử dụng như sau:

`vector<kiểu> A;`

Sắp xếp theo thứ tự tăng dần:

`sort(A.begin(), A.end());`

Sắp xếp theo thứ tự giảm dần:

`sort(A.begin(), A.end(), greater<kiểu>());`

***Yêu cầu sinh viên tìm hiểu các biến thể của các hàm `find()` và `sort()` trong slide tham khảo STL.***