**2.VECTOR**

**a.Nhập xuất vector.**

#include <bits/stdc++.h>

#define int int\_fast64\_t

using namespace std;

main()

{

int n;

cin >> n;

vector<int> a;

for (int i = 0; i < n; ++i)

{

int x;

cin >> x ;

a.push\_back(x);

}

for (int i = 0; i < a.size(); i++) cout<<a[i]<<" ";

//hoac

for (int i : a) cout<<i<<" ";

return 0;

}

**b. Xoá phần tử, vị trí.**

//Xoa phan tu tai vi tri m

int m;

cin>>m;

a.erase(a.begin() + m);

for (int i : a) cout<<i<<" ";

// Xoa phan tu co gia tri bang k trong vector

int k;

cin>>k;

a.erase(remove(a.begin(), a.end(), k), a.end());

for (int i : a) cout<<i<<" ";

**c. Chèn phần tử trong vector**

// Chen phan tu s vao vi tri t

int s,t;

cin>>s>>t;

a.insert(a.begin() +s, t);

for (int i : a) cout<<i<<" ";

**d. Tìm giá trị lớn nhất nhỏ nhất trong vector**

//Tim Min, Max

int mina=\*min\_element(a.begin(), a.end());

int maxa=\*min\_element(a.begin(), a.end());

**e. Tìm vị trí giá trị lớn nhất nhỏ nhất trong vector**

//Tim vi tri Min, Max

auto x=min\_element(a.begin(), a.end());

auto y=max\_element(a.begin(), a.end());

int vtmin= distance(a.begin(), x);

int vtmax= distance(a.begin(), x);

**k. Tìm kiếm tuần tự**

auto it=find(a.begin(),a.end(),pt);

if(it!=a.end())

vt=distance(a.begin(),it);

cout<<vt;

**f. Sắp xếp vector**

//Sap xep vector

sort(a.begin(), a.end());

**g.Tìm kiếm nhị phân**

int value;

cin>>value;

cout<< binary\_search(a.begin(), a.end(), value);

**h. Tìm giá trị đầu tiên lớn hơn hoặc bằng value**

int value;

cin>>value;

auto it = lower\_bound(a.begin(), a.end(), value);

//Giá trị:

cout<<\*it<<endl;

//Vị trí

cout<<distance(a.begin(), it);

**i. Tìm giá trị đầu tiên lớn hơn value**

int value;

cin>>value;

auto it = upper\_bound(a.begin(), a.end(), value);

cout<<\*it<<endl;

cout<<distance(a.begin(), it);