**Bài 1 : Vào ra trong C++.**

#include<iostream>

#include<iomanip>

using namespace std; // sử dụng hàm sau này .

int main(){

int Bankinh; // khai báo biến

int a=100,b=200,c=300;

cout << a << " " << b << " " << c << endl; // in ra màn hình

cin >> a; // nhap gia tri a tu ban phim

cout<< a << endl; // in ra màn hình

cout << “ Xin chao” << endl; // thêm endl để xuống dòng ;

cout << " Xin chao !\n"; // in ra màn hình

float d;

cin >> d;

cout << fixed << setprecision(4) << d << endl; // in ra do chinh xac thêm thu vien moi là include < iomanip >

bool ok = true;

cout << ok << endl; // 1 = true , 0 = false;

char ktu;

char ktu = 'r' ;// nhay don

cin >> ktu;

cout<< ktu << ;

return 0;

}

\*Kiểu dữ liệu :

- Kiểu dữ liệu số: n bit 🡪 mất 1 bit làm bit dấu 🡪 số nguyên có thể làm được :

+ số nguyên : int ( 4byte = 32 bit) , long long ( 8 byte = 64 bit), unsigned int : biểu diễn không dấu , unsigned long long : biểu diễn không dấu , short , char.

+số thực : float ( 4byte , 6 số sau dấu thập phân ) , double ( 8byte , 14 số sau dấu thập phân ) .

* Kiểu dữ liệu khác:

+ Kí tự : char ( 1 byte : 256 giá trị : 0 🡪 255 )

+ Đúng sai : bool ( 2 giá trị true , false ) 1 byte .

\*Biến :

Cthuc : Kieu\_Du\_Lieu Ten\_Bien ;

Qui tắc :

- Không được bắt đầu bằng chữ số ;

- Không được có dấu cách ;

- Không được trùng với các keyword.

\*Dấu  // cmt , giải thích , chú thích .

**Bài 2 : Toàn tập về các toán tử cơ bản trong C++.**

#include <iostream>

using namespace std;

int main() {

int a;

cout << " Xin chao !\n"; // in ra dong chu xin chao

cin >> a; //nhap gia tri cho bien a

cout << a << endl; // in ra gia tri a

a=200; // toan tu gan

cout << a << endl;

// toan tu toan hoc + , - , \* , / , % ( lay du )

a = 100 ;

int b =200;

int c = a+b; // c=300;

cout << c << endl;

float thuong = (float) a/b;

// float thuong = 1.0 \* a/b;

cout << thuong << endl;

a++; // tang sau

// ++a; tang truoc

// vd a = 100 , int b = a++ thì in ra a= 101 và b = 100

// a = a+b tuong tu a += b ;

// toan tu so sanh : < > <= >= == !=

// toan tu logic : AND(&&) OR(||) NOT(!(menhde))

return 0;

}

**Bài 3 : Cấu trúc rẽ nhánh trong ngôn ngữ lập trình C++ IF ELSE và SWITCH CASE**

* Một số thư viện :

+ #include <math.h> :

. abs(x) giá trị tuyệt đối

. pow(a,b) Tính a^b : trả về số double

. sqrt(n) : tính căn bậc 2 của n : trả về số double

. round(a) : làm tròn số thập phân gần nhất

* Cấu trúc IF ELSE , SWICH CASE

#include <iostream>

using namespace std;

int main(){

if(100>50){

cout << "YES" << endl;

}

else cout << " NO " << endl;

int n ;

cin >> n ;

if (n) { // 0 : false , != 0 true

cout << "YES" << endl;

}

int day=4;

switch (day) {

case 1:

cout << "Monday" << endl;

break;

case 2: // tuong tu else if

cout << "Tuesday" << endl;

break;

case 3 : case 4 : case 5 : // gop nhieu case

cout << " HELOO ";

break;

default : // tuong tu else

cout << "du lieu khong hop le";

}

return 0;

}

**Bài 4 : Vòng lặp For , WHILE , DO WHILE**

#include <iostream>

using namespace std;

int main(){

for( int i = 1 ; i <= 10 ;i++){ // tuong tu C , 1 trong 3 statement co the khuyet

cout << " Hello \n";

}

// su dung bien i trong for thi bien ay chi thuoc pham vi vong for;

// break vong lap se dung luon tai do

// continue vong lap se ko thuc hien code o duoi nua , nhay sang vong lap khac

int i=0;

while ( i<5){

cout << i << endl;

++i;

}

i=0;

do {

cout << i << endl;

++i;

}

while (i<10);

return 0;

}

**Bài 5 : Hàm , Nạp chồng hàm , tham chiếu và tham trị .**

* Hàm void tương tự như C .
* Hàm không làm thay đổi giá trị của 1 biến a
* Tham chiếu , tham trị như C
* Dùng bool để return true , return false thay cho return 1 , return 0 như C
* Nếu hàm b có chứa hàm a thì hàm a phải khai báo trước hàm b để tránh lỗi.
* Khai báo nguyên mẫu hàm để tránh lỗi : bool nt(int) ; bool check(int) ; int min(int,int) ( khai báo nguyên mẫu hàm a và b luôn )
* Cách dùng tham chiếu trong C++ : thêm & trước đối số

Int tang( int &n){

++n;

}

Int main(){

Int a = 100;

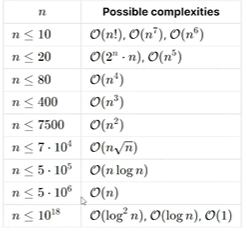
Tang(a);

Cout << a << endl; // a đã đc tăng lên và in ra 125;

}

* Function overloading nếu các hàm cùng tên mà khác dữ liệu biến thì chạy ctr thì tự khắc chọn hàm cho phù hợp

**Bài 6 : Phân tích độ phức tạp của thuật toán .**

* Big O Notation .
* Độ phức tạp .
* ****

**Bài 7 : Mảng 1 chiều trong C++.**

#include <iostream>

using namespace std;

// Cach khai bao : Kieudulieu tenmang[soluongphantucuamang]

int main(){

int a[100]; // mang tinhx

int b[10] = {10,9,8,7,6,5,4,3,2,1};

// chi so cua mang n chay tu 0 -> n-1.

cout << b[2] << endl;

int n;

cin >> n;

int c[n];

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

cin>>a[i]; // nhap gia tri cho phan tu o vi tri i

}

// cach in tuong tu nhu ngon ngu C

int d[10]={0} // khai bao mang co tat ca cac phan tu = 0

// cach duyet nhanh for each

for ( int x:b){

cout << x <<" ";

}

return 0;

}

**Bài 8 : Mảng 2 chiều và các bài toán quan trọng trong C++**

#include <bits/stdc++.h>

using namespace std;

void nhap(int a[][100] , int n , int m ){

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

for(int j =0 ; j<n ; j++){

cin >> a[i][j];

}

}

}

int main(){

int a[5][10] // 5 la sohang , 10 la socot va gom 50 phan tu.

int b[5][10]= {(1,2,4),(4,5,6)};

int n,m;

cin >> n >> m;

int c[n][m];

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

for(int j =0 ; j<n ; j++){

cin >> c[i][j];

}

}

// in tuong tu nhu ngon ngu C .

int d[100][100];

nhap(d,n,m);

// trong C++ khong doc dau cach

// khai bao mang cac vecto

int n,m ; cin>>n;

vertor<int> e[n];

//1 2

// 2 3 4

// dung vecto moi dong co the co cac ptu khac nhau

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

int m ; cin>>m; // so luong phan tu tren dong do

for(int j = 0;j<m;j++){

int x; cin >> x; // nhap 1 thang can

a[i].push\_back(x);

}

}

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

for(int j=0;j<a[i].size();j++){

cout << a[i][j]<< " ";

}

cout << endl;

}

// hoac dung string

int n,m ; cin >> n;

string a[n];

cin.ignore();

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

getline(cin,a[i]);

}

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

cout << a[i] << endl;

}

}

**Note : Mảng cộng dồn trên mảng 1 chiều và mảng 2 chiều // truy vấn tổng trên đoạn**

#include <bits/stdc++.h>

using namespace std;

int main(){

// tinh tong tu vi tri left -> right

// lam binh thuong mat O(n)

int n;

cin>> n;

int a[n];

for(int &x:a) cin>> x;

int q;

cin >> q;

while(q--){

int l,r;

cin >> l >> r;

int sum=0;

for(int i=l;i<=r;i++){

sum+=a[i];

}

cout << sum << endl;

}

// dung Prefix Sum co O(1)

// pre[k]=tong tu 1 -> n

//[l,r]: pre[r]-pre[l-1]

int n; cin >> n;

int a[n+1];

for(int i=1;i<=n;i++) cin>>a[i];

int prefix[n+1]={0};

for(int i=1;i<=n;i++){

prefix[i]=prefix[i-1]+a[i];

}

int q; cin>>q;

while(q--){

int l,r;

cin >> l >> r;

cout << prefix[r]-prefix[l-1] << endl;

}

// Prefix Sums tren mang 2 chieu : tinh tong theo hinh chu nhat

// cach thong thuong

int n,m;

cin>>n>>m;

int a[n+1][m+1];

for(int i=1;i<=n;i++){

for(int j=1;j<=m;j++){

cin>>a[i][j];

}

}

int q;

cin>>q;

while(q--){

int h1,h2,c1,c2;

int sum=0;

cin >> h1 >> h2 >> c1 >> c2;

for(int i=h1;i<=h2;i++){

for(int j=c1;j<=c2;j++){

sum +=a[i][j];

}

}

cout << sum << endl;

}

// cach 2 : prefix[i][j] : tong tat ca phan tu cua hcn bat dau tu hang 1 den hang i , cot 1 den cot j

//prefix[i][j]=prefix[i-1][j]+prefix[i][j-1]-prefix[i-1][j-1]+arr[i][j];

// sum = prefix[A][B]-prefix[a-1][B]-prefix[A][b-1]+prefix[a-1][b-1] vs i tu a den A va j tu b den B

int n,m;

cin>>n>>m;

int a[n+1][m+1];

for(int i=1;i<=n;i++){

for(int j=1;j<=m;j++){

cin>>a[i][j];

}

}

int prefix[n+1][m+1]={0};

//memset(prefix,0,sizeof(pre));

for(int i=1;i<=n;i++){

for(int j=1;j<=m;j++){

prefix[i][j]=prefix[i-1][j]+prefix[i][j-1]-prefix[i-1][j-1]+a[i][j];

}

}

int q;

cin>>q;

while(q--){

int h1,h2,c1,c2;

int sum=0;

cin >> h1 >> h2 >> c1 >> c2;

sum = prefix[h2][c2]-prefix[h1-1][c2]-prefix[h2][c1-1]+prefix[h1-1][c1-1];

cout << sum << endl;

}

}

**Bài 9 : Xâu kí tự STRING.**

#include <bits/stdc++.h>

using namespace std;

int main(){

string s;// cach khai bao

cin >> s;

cout << s << endl;

string d = "Python"; // cach khai bao

// string ko nhan dau cach

// muon doc dau cach dung getline

cin.ignore(); // de xoa enter trong cin

getline (cin,s); // giong gets trong C bi anh huong boi scanf() (hay cin>> )

cout <<s << endl;

// truy cap tung ki tu vao chuoi tuong tu mang

//length() : chi ra do dai cua sau

cout << s.size() << endl;

cout << s.length() << endl;

// hai ham tren deu chi ra do dai cua xau ki tu

// cthuc : xaukitu.length()

// Muon duyet qua tung ki tu dung nhu sau:

for(int i=0;i< s.length());i++{

cout << s[i] << endl;

}

// hoac dung fot(char x:)

for(char x : s){

cout << x << endl;

}

// noi hai xau dung toan tu cong

string a = "python";

string b = "java";

a+=b;

string c = a+b;

cout << a << " " << c<< endl; // xau c bang xau a noi xau b ;

//so sanh hai tu theo thu tu tu dien

string a = "abc";

string b = "def";

if(a>b){

cout << "a lon hon b";

}

if(a==b){

cout << " a bang b ";

}

else{

cout << "a nho hon b ";

}

// hoac so sanh bang ham compare

// a < b : -1;

// a==b : 0;

// a>b : 1;

cout << a.compare(b) << endl;

// muon cat xau kitu dung ham substr(vitribatdau,sokitucancat)

string a = "abcdef";

string b= a.substr(2,3);

// chuyen xau thanh so tu nhien dung ham stoi chuyen thanh interger

string a = "123456";

int x = stoi(a);

cout << x << endl;

// neu so qua lon se bi tran thi se dung stoll(a) chuyen thanh lontrig long

long long x = stoll(a);

// ham bien so nguyen thanh 1 chuoi dung ham to\_string

int n = 123456;

string s = to\_string(n);

cout << s << endl;

// trong C++ chap nhan mang char trong C.

// tach tung tu trong xau ki tu dung stringstream

string s = "java python php lap trinh";

stringstream ss(s); // tach theo dau cach

string tmp;

while(ss>>tmp){ // tach theo dau cach

cout << tmp << endl;

}

// muon tach theo dau cham

while(getline(ss,tmp,'.')){

cout << tmp << endl;

}

// Thu vien can dung tu gio la #include <bits/stdc++.h>

}

**Bài 10 : Dùng VECTOR trong mảng**

#include<bits/stdc+.h>

#include <vector>

using namespace std;

int main(){

// Vecto nhu 1 mang dong khong can 1 kich thuoc co dinh

// vecto<kieu\_du\_lieu> ten\_vecto

vecto<int> v;

// ham push\_back() day 1 phan tu vao vecto

v.push\_back(10);

v.push\_back(20);

// cu moi lan them vao thi phan tu moi duoc them vao cuoi vecto

// ham size() : kich thuoc vecto

cout << v.size() << endl;

// co the truy cap theo chi so

cout << "Phan tu dau tien trong vecctor : "<< v[0] << endl;

cout << "Phan tu cuoi cung trong vector : "<< v[v.size()-1] << endl;

cout << "Phan tu cuoi cung trong vector : "<< v.back() << endl;

// Duyet theo chi so

.; . Duyet thong qua index , for each.

for(int i = 0 ; i<v.size();i++){

cout << v[i] << endl;

}

for(int x:v){

cout << x << endl;

}

// v.end() : phan tu sau phan tu cuoi cung

// Duyet theo iterator : con tro thong minh

for(vector<int>::iterator it = v.begin();it!= v.end;++it){

cout << \*it << endl;

}

// Duyet theo auto : thay cho so , string

for(auto it = v.begin();it!= v.end(); ++it){

cout<<\*it<<endl;

}

cout << v[2] << endl;

// hoac cung thong qua \*(v.begin()+index)

cout << \*(v.begin()+2)<<endl;

int n ; cin >> n;

vector<int> v(n) // khai bao vecto nhung da co n phan tu ;

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

cin >> v[i];

}

// hoac cach khac nhu sau : dung bien tam

int n ; cin >> n;

vector<int> v;

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

int x;

cin>> x;

v.push\_back(x);

}

// hoac khoi tao gia tri cho vecto

int n;

cin>>n;

vecto<int> v(n,100);

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

cout << v[i] << endl;

}

v.push\_back(20); // van duoc noi rong vector ra duoc

// dung cho ca long long va string nhu int

// hoac vecto trong vecto luon

// Trong string ta cung co the dung vector de tach

string s =" Ngon ngu lap trinh "

stringstream ss(s);

string tmp;

vector<string> v;

while(ss>>tmp){

v.push\_back(tmp);

}

for(string x:v){

cout << x << endl;

}

}

**Bài 11: Pair trong C++**

#include<bits/stdc+.h>

using namespace std;

int main(){

// pair : dung de luu cac cap so vs so , chuoi vs so ,..

// pair<datatype1,datatype2> v;

pair <int,int> v;

pair <long long , long long > v;

pair <string , int > v;

// truy cap qua v.first va v.second

// khoi tao gia tri dung make\_pair

pair<int,int> v = make\_pair(100,200);

// C++11 ko can su dung make\_pair

pair<int,int> v = {100,200};

// tuple de dung luu 3 so , 4 so;

tuple<int,int,int> t(1,2,3);

// truy cap qua chi so

cout << get<0>(t) << endl;

// hoac dung make\_tuple

tuple<string,int,int> t = make\_tuple("java",100,2);

}

**Bài 12: Cấu trúc dữ liệu Set trong C++**

#include <bits/stdc+.h>

#include <set>

using namespace std;

//O(logn)

int main(){

set<int> s; // khai bao

s.insert(100);

s.insert(200);

s.insert(100);

cout << s.size() <<endl;

// set chi chua nhung phan tu khac nhau , ko chua phan tu trung nhau

// set luon xep cac phan tu theo thu tu tu nho toi lon

// Dung find , count de tim kiem

set<int> s;

for(int i = 0 ; i<10 ;i++){

s.insert(i);

}

if(s.count(5)!=0){

cout << "Found";

}

if(s.find(5)!=s.end){

cout << "Found" << endl;

}

// ham Erase : xoa phan tu

s.erase(5);

cout << s.size() << endl;

// duyet phan tu

for (int x:s){

cout << x << " ";

}

cout << endl;

s.erase(s.find(5)); // dung interator se ko xoa het tat ca phan tu trong multiset

cout <<\*s.begin()<< endl; // ptu dau

cout <<\*s.rbegin()<< endl;// ptu cuoi

for (auto x : s ){ // auto dung thoai mai

cout<<x<<endl;

}

for(set<int>::iterator it= s.begin();it!=s.end();++it){

cout << \*it << endl;

}

// muon luu cac tap hop luu Cac phan tu giong nhau thi dung multiset<int> ms;

multiset<int> ms;

ms.insert(100);

ms.insert(200);

ms.insert(100);

cout << ms.size() <<endl;

// van luu theo thu tu tu nho toi lon

ms.erase(100); // xoa het cac phan tu 100 trong tap hop

for(int x:ms){

cout << x << endl;

}

// unordered\_set : luu cac phan tu rieng biet nhung ko xep theo thu tu nua

unordered\_set<int> s;

for(int i;i<10;i++){

s.insert(i);

}

for(int x:s){

cout << x << endl;

}

return 0;

}

**Bài 13 : Cấu trúc dữ liệu map trong C++**

#include <bits/stdc+.h>

using namespace std;

int main(){

//map<datatype1,datatype2> mp;

//(key,value)

//map luu cac pair

map<int,int> mp;

mp[100]=200; //(100,200)

mp[200]=300;//(200,300);

mp.insert({300,400}); // (300,400);

cout << mp.size() <<endl;

mp[100]=300; //(100,300)

cout<< mp.size() << endl;// thay doi gia tri key , ko thay doi nhu set

// duyet qua map

for(pair<int,int> x : mp){

cout << x.first << " " << x.second << endl;

}

for(auto it:mp){

cout << it.first << " " << it.second << endl;

}

for(map<int,int>::interator it=mp.begin();it != mp.end();it++){

cout<<(\*it).first <<" "<<(\*it).second << endl;

}

// tim kiem dung count hoac find

if(mp.cout(100)!=0){

cout <<"Found" <<endl;

}

if(mp.find(100)!=mp.end()){

cout <<"Found" <<endl;

}

// erase : xoa ca cap

mp.erase(100);

for(auto it:mp){

cout << it.first << " " << it.second << endl;

}

// khi truy cap map chua ton tai thi luon bang 0

// map cung sap sep tang dan theo key

// multimap va unordered\_map tuong tu nhu set

multimap<int,int> mp;

// dung multimap ko dung duoc phep gan

mp.insert({1,100});

mp.insert({2,200});

mp.insert({1,200});

for(auto it:mp){

cout << it.first << " " << it.second << endl;

}

}

**Bài 14 : Thuật toán tìm kiếm nhị phân hàm Lower\_bound và Upper\_bound.**

#include <bits/stdc+.h>

#include <algorithm>

using namespace std;

bool ls(int a[],int n,int x){ // Linear Search

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

if(a[i]==x)

return true;

}

return false ;

}

bool bs(int a[],int n,int x){// Binary Search

int l=0,r=n-1;

while(l<=r){

int m =(l+r)/2 // l + (r-l)/2

if(a[m]==x){

return true;

}

else if(a[m]<x){

l=m+1;

}

else{

r=m-1;

}

}

return false;

}

bool binary\_search(int a[],int l , int r , int x ){ // dung de quy

if(l>r) return false;

int m = (l+r)/2;

if(a[m]==x) return true;

else if (a[m]<x) return binary\_search(a,m+1,r,x);

else return binary\_search (a,l,m-1,x);

}

// ham co san binary\_search(a+x,a+y,key) --> a[x] -> a[y-1]

// binary\_search(v.begin()+x,v.begin()+y,key)--> v[x]->v[y-1] -> dung cho vector

// binary\_search(v.begin(),v.end(),key)

// tim vi tri phan tu dau tien cua 1 phan tu x trong 1 mang da sap xep

int first\_pos(int a[],int n , int x){

int res = -1;

int l =0 , r = n-1;

while(l<=r){

int m = (l+r)/2;

if(a[m]==x){

res = m;

r=m-1;

}

else if(a[m]<x){

l=m+1;

}

else r = m-1;

}

return res ;

}

// tim vi tri phan tu cuoi cung cua 1 phan tu x trong 1 mang da sap xep

int first\_pos(int a[],int n , int x){

int res = -1;

int l =0 , r = n-1;

while(l<=r){

int m = (l+r)/2;

if(a[m]==x){

res = m;

l=m+1;

}

else if(a[m]<x){

l=m+1;

}

else r = m-1;

}

return res ;

}

// Ap dung cho mang , vecto duoc sap xep , set , map cua 2 ham duoi nhu sau :

// ham lower\_bound(iter1,iter2,key) tim [iter1,iter2) -> Tra ve vi tri cua phan tu dau tien >= key

// dung auto cho de

// Neu ma tat ca cac phan tu trong mang deu nho hon key -> tra ve iter2

// ham upper\_bound(iter1,iter2,key) tim [iter1,iter2) -> tra ve vi tri dau tien cua phan tu > key

// Neu tat ca cac phan tu trong mang deu nho hon key -> tra ve iter 2;

int main(){

//thuat toan tim kiem tuyen tinh Linear Search

// O(n)

// thuat toan tim kiem nhi phan Binary Search

// O(logn)

// dk : Cac phan tu phai duoc sap xep tang dan , giam dan

// lower\_bound

int n , x;

cin >> n >> x;

int a[n];

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

cin >> a[i];

}

auto it = lower\_bound(a,a+n,x); // tim vi tri dau tien cua phan tu >= x nam trong mang a

cout << \*it << endl; // tra gia tri

cout << it-a <<endl; // tra vi tri

}

**Bai 15 : Thuat toan sap xep sort va stable\_sort**

#include <bits/stdc+.h>

#include <algorithm>

using namespace std;

// sort

// stable\_sort

// cach viet comparator

//bool cmp(int a , int b){ // sap xep giam dan

// if(a>b) return true; // a se dung truoc b

// else return false;

//}

bool cmp(int a , int b){

return abs(a)>abs(b);

}

int main(){

int n ;

cin >> n;

int a[1000];

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

cin>>a[i];

}

sort(a,a+n); // ham sap xep a[x] -> a[y] sort(a+x,a+y+1)

// se sap xep theo thu tu tang dan

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

cout << a[i] << " ";

}

sort(a,a+n,greater<int>()) // se sep theo thu tu giam dan

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

// dung greater<kieudulieu> de sap xep giam

// ham sort co the xep ki tu trong xau , va sap xep cac xau

// ham sort sap xep pair thi theo thu tu first tang dan

// stable\_sort : bubble , insertion sort , merge sort : tinh on dinh cua thuat toan sap xep

// vd : sap xep cac phan tu trong mang theo thu tu gia tri tuyet doi tang dan

// Input : 1 -1 2 1 5 -2 -5

// Output : 1 -1 1 2 -2 5 -5

int n;

cin >> n;

int a[n];

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

cin >> a[i];

}

stable\_sort(a,a+n,cmp);

for(int x:a){

cout << x << " ";

}

}

# \*số hoàn hảo : nếu p là số nguyên tố và là số nguyên tố thì ( là số hoàn hảo.

Cấu trúc hàm insert

#include <bits/stdc++.h>

using namespace std;

int main(){

ios\_base::sync\_with\_stdio(0);

cin.tie(0);

string a="nguyentheduong";

string b="ptit";

cout <<"Xau a: " <<a << endl;

cout <<"Xau b: " <<b<< endl;

cout << endl;

int s=2;

cout << "Lan 1 " << endl;

cout << "Gan xau b vao vi tri dau cua xau a va gan cho xau c \n" << endl;

string c = a.insert(0,b);

cout <<"Xau c gan : " << c << endl;

cout <<"Xau a : " << a << endl;

cout << "Voi ham a.insert(vitri,xaukitu) thi gia tri cua ham a ko doi " << endl;

cout << "\n\nLan 2" << endl;

cout << "Gan 2 ki tu chu a vo vi tri dau\n" << endl;

string d = a.insert(0,s,'a');

cout <<"Xau d gan : " << d << endl;

cout <<"Xau a : " << a << endl;

cout << "Voi ham a.insert(vitri,soluong,kitu) ko the insert cho xaukitu thi gia tri cua ham a se thay doi theo " << endl;

cout << "\n\nLan 3 :" << endl;

string e = a.insert(0,s+1,'3');

cout <<" Gan 3 ki tu so 3 vao vi tri dau vao xau e \n" << endl;

cout <<"Xau e : " << e << endl;

cout <<"Xau c : " << c << endl;

cout <<"Xau a : " << a << endl;

}

4 27

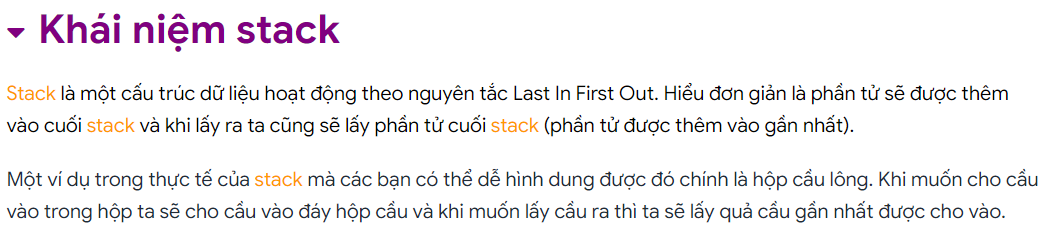
A 999

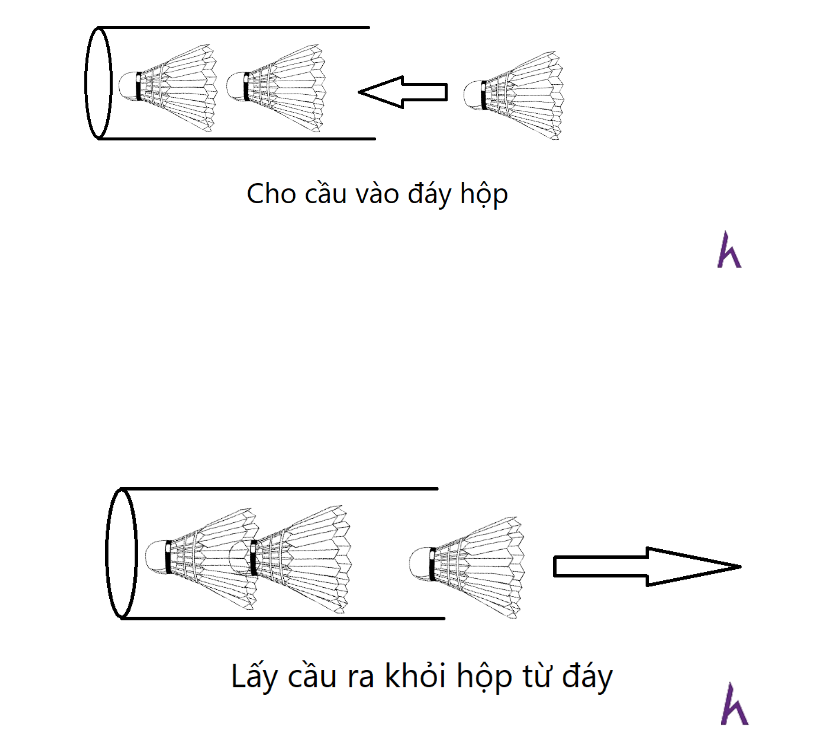
B 999;

S=0;

### **Bài 16 : Stack và queue**

1. **Stack**

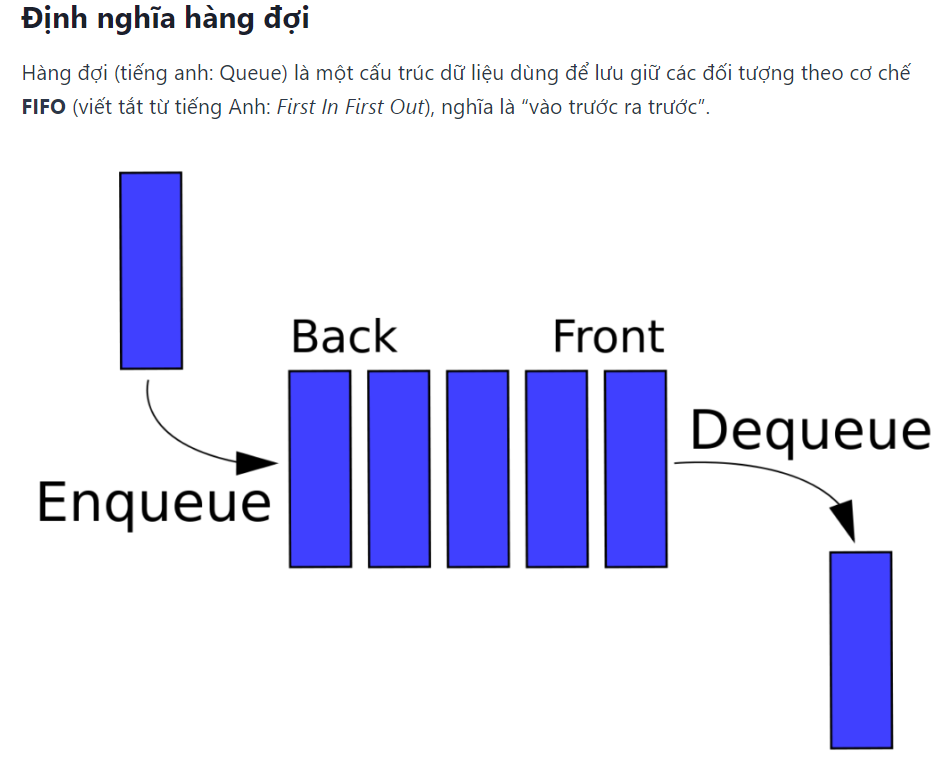
****

****

Các phương thức cơ bản trong stack của C++:

* **push**: Thêm phần tử vào cuối stack
* **pop**: Loại bỏ phần tử cuối stack
* **top**: Trả về giá trị là phần tử cuối trong stack
* **size**: Trả về giá trị nguyên là số phần tử đang có trong stack
* **empty**: Trả về một giá trị bool, true nếu stack rỗng, false nếu stack không rỗng

1. **Hàng đợi**



Một cấu trúc queue có các chức năng cơ bản như sau:

* **Dequeue:**Lấy ra phần tử đầu queue.
* **Enqueue:**Thêm một phần tử vào cuối của queue.
* **IsEmpty:**Kiểm tra xem queue hiện tại có đang rỗng hay không.
* **Front:**Truy nhập phần tử ở đầu queue.

Trong C++ STL các chức năng của Queue là gì:

* **empty():** Mục đích là để Kiểm tra hàng đợi hiện tại có đang rỗng không.
* **size():** Là để trả về kích thước hàng đợi hiện tại.
* **front():** Dùng để trả về phần tử đầu tiên của Queue.
* **back():** Dùng để trả về phần tử cuối cùng của hàng đợi.
* **push():** Để nạp thêm một phần tử vào Queue, biến nạp thêm phải cần được khởi tạo trước đó.
* **emplace():** Để nạp thêm một phần tử vào Queue, biến có thể được khởi tạo ngay tại thời điểm nạp.
* **pop():** Dùng để Xóa một phần tử ở đầu hàng đợi.
* **swap():** Sử dụng khi cần hoán đổi nội dung giữa 2 hàng đợi