بسم الله الرحمن الرحيم

STL

1-vector :

علشان تحول من ال array ل vector

Int arr[6]={1,2,3,4,5,6];

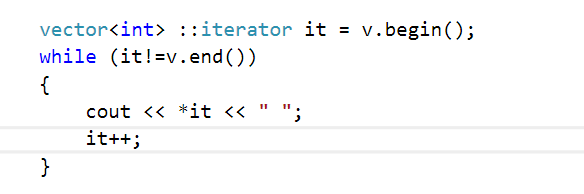
Vector<int> v1(arr,arr+6);

في خاطية كمان ف ال vector انك تقددر تجيب اول و اخر عنصر ع طول

v.front()--------v.back();

علشان طبع ال vector في شكل عملي علشان تبع اي حاجة ليها بداية ونهاية

ودي الطريقة ال أمثل للعملية دي



دلوقت ف النفرض انك معاك vector وعاوز تعملو sort بس بالعكس يعني

عاوزين نتربة من الاكبر

sort(v.rbegin(), v.rend());

2d vector<int>

لو عاوزين نعمل مثلا array 2d ونخلي كل العناصر ال جواها (5)

vector< vector<int> > v2d(10, vector<int>(20,5));

لو حبيت بقي تعمل erase لعنصر ف ال vector هي بتبقي شبه ال string بس مش اوي

بمعني

vector<int>v1(5, 7);

//حذفت عنصر من ال vector

v1.erase(v1.begin() + 1);

// create ilment in indix 2 = 5 ;

v1.insert(v1.begin() + 2,5);

بنفس المنظر ال فوق دا العملية حلوة اهي ^^

Keep going

طب لو عاوز نفس ال فيكتور ال فوق دا بس هضيف علية عنصر ف الأخر لو فاكر ان الكلام دا كا مستحيل ف ال array لانها كانت standard يعني مينفعش العب فيها اما ال vector لا يا حبيبي دي حاجة cute كدا

يلي نشوف ازاي

v1.push\_back(20);

بس كدا بالبساطة دي هو هيروح لأخر عنصر ف ال vector ويضفلك 20

لو عاوز check ع عنصر موجد ف ال vector ولا لا

if (find(v1.begin(), v1.end(), 3) != v1.end()){

cout << true;

}

else cout << false;

functions for vector very important

دي بتجبلك المسافة ما بين ال عناصر وخد بالك من حتة السالب دي

cout << distance(v1.begin(), v1.begin() + 3); //3

cout << distance(v1.begin()+4, v1.begin()); // -5

لو عاوزين ندبل قيمة ب قيمة تاني ف ال vector بس بشرط انك تكون متأكد ان القيمة موجودة فعلا

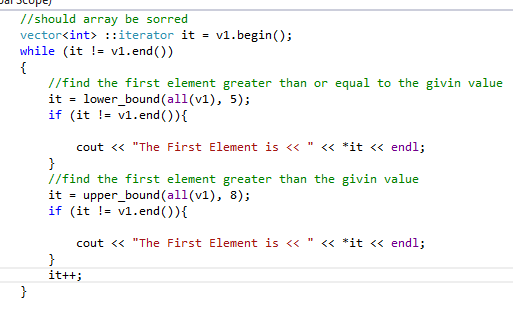
replace(all(v1), 7, 9);

ممكن بقي نخلي موضوع ال replace دا اكثر دقة

بمعني اقلو كل ما تلاقي حاجة معينة استبدلها بحاج تاني يعني طبقا للشرط دا

replace\_if(all(v1), "شرط معيين ",51)

خلينا بقي نتعلم حاجة جديدة

دي بقي حاجة هستعملها كتير بعد كدا واظن شرحها واضح فوق دوون داعي للكلام

دلوقت بقي لازم نتعلم حاجة جديدة بالنسبة لل max و min

انت ف كل مرة علشان تجيب ال قيمة ال max بتولع الدنيا علشان تجيبها

تعالي نختصرها ف سطر واحد

int mx = \*max\_element(all(v1)); cout << mx << " max " << endl;

int mn = \*min\_element(all(v1)); cout << mn << " min " << endl;

بس كدا شوفت ال بساطة حلوة ازاي ...

تعالي بقي للحلو

swap\_ranges(v1.begin(), v1.begin() + 2,v2.begin());

دي بقي يا سيدي بتقلو اعمل swap من كذا ل كذا ع اساس كذا

يارب تكون وصلت هنشتغل عليها كتير الفترة ال جاية بإذن الله فوق كدا يا حبيبي وروق

دلوقت بقي لو عاوزين نوجد جمع ال فيكتور او ضرب ال فيكتور من غير تضييع وقت

int vecsum = accumulate(all(v1), 0);

int vecmul = accumulate(all(v1), 1,multiplies<int>());

cout << vecsum << endl << vecmul << endl;

في بقي مسألة خزوقك فيها حج مؤمن اسمها perimination

ودي شرحهاف السريع والنفرض معاك array فيها

4 3 1

ف دي هتديك 6 حالات اي عدد حدوث الاحتمالات دي

1 3 4

1 4 3

3 1 4

3 4 1

4 1 3

4 3 1

sort(all(v2));

int c = 0;

do{

vector<int> ::iterator t = v1.begin();

while (t!=v1.end())

{

cout << \*t << " ";

t++;

}

c++;

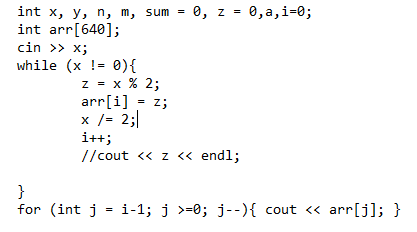
cout << endl;

} while (next\_permutation(all(v1)));

cout << c << " " ;

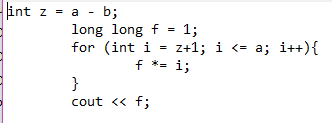
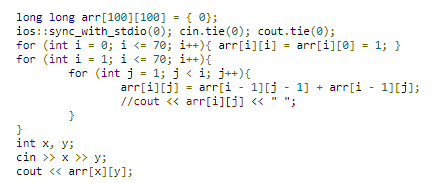
شوية اكواد جاهزة بقي لوزم الروشنة !!

تحويل من decimal ل binary

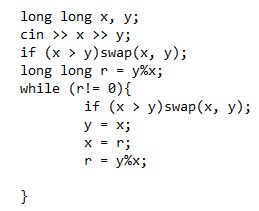


كود توافيق !!

دا كود تباديل !!



GCD



خد التركة دي هتنفعك



6C:88:14:38:62:E4

Reference level 2 cp

Sort nlog n;

في 3 function لأازم لما تستخدمهم تبقي عامل سورت

Binary\_search(num)

Lower\_bound ترجع اول قيمة تساوويني او اكبر مني

Upper\_bound ترجع اول قيمة اكبر مني

Fill تملا عناصر اراي او فيكتور