<algorithm>

Count\_if (begin.pointer(),last.pointer(),function) – возвращает кол-во элементов удовлетворяющих

function (begin/last начало/конец массива,вектора

Function возвращает значение True/False)

Bool f(int x){if (x%2==0)return true; return false;}

Int cnt=count\_if(a.begin(),a.end(),f);

Rotate(begin,middle,last) – до middle (исключая) перенос в правую сторону, от middle (включая) до

last сдвигает в левую сторону

1 2 3 4 5 6 7 (а+3)ий элемент = 4

Rotate(a,a+3,a+7); ответ: 4 5 6 7 1 2 3

1 2 3 4 5 6 7 8 9 10

Rotate(1,3,1); ответ: 1 4 5 6 7 8 9 2 3 10

Fill (begin,end,fill-parameter); - начиная с begin заполняет параметром до end элемента (исключая)

a(19); \\ обязательно передавать количество элементов

fill(a.begin(),a.begin()+3,3);

fill (a.begin()+3,a.end(),4);

Unique (begin,last) -удаляет одинаковые подряд идущие элементы кроме одного, возвращает

указатель на элемент после последнего

1 1 1 2 2 2 3 3 2 2 1 1 -> 1 2 3 2 1 \_

| |

It1 unique it2

Unique(begin, last, function) – удаляет по функции

sizeof(a); - количество байтов выделенных на массив

n=sizeof(a)/sizeof(int); - размерность массива

|  |
| --- |
| #include <iostream> |
| #include <algorithm> | |
| #include <vector> | |
|  | |
| using namespace std; | |
| //int a[] = {1, 1, 1, 2, 2, 2, 3, 3, 3, 2, 2, 1, 1}; | |
| int a[] = {1, 2, 3, 1, 2, 3, 4}; | |
|  | |
| // bool f(int a, int b) { | |
| // return a > b; | |
| // } | |
|  | |
| int main() { | |
| int n = sizeof(a) / sizeof(int); | |
| vector<int> b(a, a + n); | |
| vector<int>::iterator it1, it2; | |
| it2 = unique(b.begin(), b.end()); | |
|  | |
| for (it1 = b.begin(); it1 != it2; it1++) | |
| cout << \*it1 << " "; | |
| return 0; | |
| } | |

For\_each (begin, end, function) – отправляет в функцию каждый элемент от begin до end

Generate (begin, end, function) – генерирует числа

Vector <int> a(10);

Int f(){ return rand()%100;}

gen (a.begin(), a.end(), f); - весь вектор заполнится случайными числами [0,99]

rand()%98+2 – случайные числа от 2 до 99

rand()%100-15 -залезет в отрицательные числа

Next\_permutation – возвращает следующую перестановку массива

123 – 132 – 213 – 231 – 312 – 321

next\_permutation (begin, end); – возвращает True/False - существует ли следующая перестановка

если массив 132, то он не выведет 123

Pred\_permutation (begin, end);

|  |
| --- |
| #include <iostream> |
| #include <algorithm> | |
| #include <vector> | |
|  | |
| using namespace std; | |
| vector<int> a; | |
| int n, x; | |
|  | |
| void f(int x) { | |
| cout << x << " "; | |
| } | |
|  | |
| int main() { | |
| cin >> n; | |
| for (int i = 0; i < n; i++) { | |
| cin >> x; | |
| a.push\_back(x); | |
| } | |
| // 1 2 3 | |
| // 1 3 2 | |
| // .. | |
| // 3 2 1 | |
| do { | |
| for\_each(a.begin(), a.end(), f); | |
| cout << endl; | |
| } while (next\_permutation(a.begin(), a.end())); | |
| return 0; | |
| } | |