# Randomized Quick sort

#include <iostream>

using namespace std;

int cnt =0;

int partition (int arr[],int i,int j){

int pivot = arr[j];

int q=i-1;

for(int p=i;p<j;p++){

cnt++;

if(arr[p] <= pivot){

q++;

swap(arr[p],arr[q]);

}

}

swap(arr[j],arr[q+1]);

return q+1;

}

int randompartition(int arr[],int i,int j){

srand(time(0));

int random = i+rand()%(j-i+1);

swap(arr[random],arr[j]);

return partition(arr,i,j);

}

void quicksort(int arr[],int i,int j){

if(i<j){

int q = randompartition(arr,i,j);

quicksort(arr,i,q-1);

quicksort(arr,q+1,j);

}

}

int main() {

int n=0;

// cin >> n;

int arr[] = {4, 3, 6, 8, 5, 9, 20, 50, 12, 30, 432, 12, 69, 29, 40, 50, 10, 4, 23, 34, 45, 45, 24,

89, 799, 45, 80, 50, 10, 30, 43,54,65,76,87, 98, 21, 32, 43, 78, 98, 57, 29, 91, 34,

54, 64, 19, 82, 87, 65};

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

// arr[i] = rand() % n;

// }

quicksort(arr,0,50);

cout <<"total cnt:"<< cnt << endl;

for(int i=0;i<51;i++){

cout << arr[i] << endl;

}

return 0;

}