#include<bits/stdc++.h>

using namespace std;

void swap(int\* a, int\* b)

/\* $$$$ THIS FUNCTION SWAPS THE POINTER TYPE VALUES

AND USED IN PROGRAM FOR SWAPPING VALUES AT THE RIGHT TIME $$$$ \*/

{

int t = \*a;

\*a = \*b;

\*b = t;

}

/\* @@@@ function takes last element as pivot,and places

the pivot at its correct position in sorted

array, and places all smaller than pivot

to left of pivot and all greater elements to right

of pivot @@@@ \*/

int partition (int arr[], int low, int high)

{

/\* LOW AS SMALLER INDEX

HIGH AS LARGEST INDEX \*/

int pivot = arr[high]; // ASSIGNING PIVOT ELEMENT

int i = (low - 1);//ASSIGNING SMALLEST INDEX

for (int j = low; j <= high- 1; j++)

{

if (arr[j] <= pivot)

{

/\* \*\*\*\* IF CURRENT ELEMENT IS SMALLER OR

EQUAL TO PIVOT ELEMENT THEN SWAP THE RESPECTIVE ELEMENTS \*\*\*\* \*/

i++;

swap(&arr[i], &arr[j]);

}

}

// SWAPPING THE PIVOT ELEMENT AT THE RIGHT POSITION

swap(&arr[i + 1], &arr[high]);

return (i + 1);

}

/\* %%%% CREATING QUICK SORT FUNCTION TO PERFORM PARTITION FUNCTION

AND SORTING THE ARRAY %%%% \*/

void quickSort(int arr[], int low, int high)

{

if (low < high)

{

int pi = partition(arr, low, high);// ASSIGNING PIVOT ELEMENT

quickSort(arr, low, pi - 1);

quickSort(arr, pi + 1, high);

}

}

int main()

{

int a;

cin>>a;

int arr[a];

for(int k=0;k<a;k++)

{

cin>>arr[k];

}

int n = sizeof(arr)/sizeof(arr[0]);

quickSort(arr, 0, n-1);

cout<<"sorted array"<<endl;

//FINALLY PRINTING THE SORTED ARRAY

for(int l=0;l<a;l++)

{

cout<<arr[l]<<" ";

}

return 0;

}