//program to perform quicksort

#include <stdio.h>

#include <stdlib.h>

void quicksort(int,int); //declaration of the quicksort and partition functions

int partition(int,int);

int arr[100];

int main()

{

int left,right,size;

int dumy[100];

printf("Enter the size of the array\n"); //accept the size of the array

scanf("%d",&size);

left=0;

right=size-1;

printf("Enter the values\n");

for(int i=0;i<size;i++)

{

scanf("%d",&arr[i]);

dumy[i]=arr[i];

}

quicksort(left,right);

printf("The entered array is :\n"); // prints the original array

for(int i=0;i<size;i++)

{

printf("%d\t",dumy[i]);

}

printf("\nThe sorted array is:\n"); // prints the new sorted array

for(int i=0;i<size;i++)

{

printf("%d\t",arr[i]);

}

return 0;

}

void quicksort(int left,int right) //quicksort function

{

if(left<right){

int pi=partition(left,right); //returns the pivot location and partitions it from there

quicksort(left,pi-1);

quicksort(pi+1,right);

}

}

int partition(int left,int right) //partition function

{

int pivot=arr[left]; //pivot is set as the leftmost element

int i=left;

int j=right;

while (i<j)

{

while(pivot>=arr[i]&&i<right)

{

i++;

}

while(pivot<arr[j])

{

j--;

}

if(i<j){

int tmp=arr[i]; // swaps the arr[i] and arr[j] elements

arr[i]=arr[j];

arr[j]=tmp;

}

}

arr[left]=arr[j];

arr[j]=pivot;

return j;

}

OUTPUT:

Enter the size of the array 10 Enter the values 3 4 2 8 6 4 9 3 1 0 The entered array is : 3 4 2 8 6 4 9 3 1 0 The sorted array is: 0 1 2 3 3 4 4 6 8 9