#include<iostream>

using namespace std;

void mergesort(int A[],int l,int r);

void merge(int A[],int l,int m ,int r);

int main()

{

int n;

cin>>n;

int A[n];

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

cin>>A[i];

int r=n-1,l=0;

mergesort(A,l,r);

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

cout<<A[i]<<" ";

int key;

cout<<endl;

cin>>key;

int i=0,j=n-1,flag=0;

while(i<=j)

{

if(A[i]+A[j]==key)

{

cout<<A[i]<<" "<<A[j]<<endl;

flag=1;

break;

}

else if(A[i]+A[j]>key)

{

j--;

}

else i++;

}

if(flag==1)

cout<<"key found\n";

}

void mergesort(int A[],int l,int r)

{

if(l<r)

{

int m=l+(r-l)/2;

mergesort(A,l,m);

mergesort(A,m+1,r);

merge(A,l,m,r);

}

}

void merge(int A[],int l,int m ,int r)

{

int n1=m-l+1;

int n2=r-m;

int L[n1],R[n2];

for(int i=0;i<n1;i++)

{

L[i]=A[l+i];

}

for(int i=0;i<n2;i++)

{

R[i]=A[m+i+1];

}

int i,j,k;

i=0,j=0,k=l;

while(i<n1 && j<n2)

{

if(L[i]>R[j])

{

A[k]=R[j];

j++;

k++;

}

else

{

A[k]=L[i];

k++;

i++;

}

}

while(i<n1)

{

A[k]=L[i];

i++;

k++;

}

while(j<n2)

{

A[k]=R[j];

j++;

k++;

}

}

Text

Description automatically generated