#include <stdio.h>

#include<unistd.h>

void mergesort();

void merge();

void quicksort();

int main()

{

pid\_t pid ;

int a[40] , n, i ;

printf("Enter the no. of integers to be sorted :");

scanf("%d", &n);

printf("Enter integers :- ");

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

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

pid = fork();

if(pid>0)

{

printf("\nInside parent process");

printf("\nProcess ID:%d" , getpid());

printf("\n\n");

quicksort(a,0,n-1);

printf("Integers sorted using quicksort : ");

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

printf("%d " , a[i]);

printf("\n");

}

else if(pid==0)

{

printf("\nInside Child process");

printf("\nProcess ID:%d" , getpid());

printf("\n\n");

mergesort(a,0,n-1);

printf("Integers sorted using mergesort : ");

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

printf("%d ",a[i]);

printf("\n");

}

else

printf("\nChild Process could not be created!\n");

return 0 ;

}

void mergesort (int a[] , int first , int last)

{

int mid ;

if (first < last)

{

mid = (first + last) / 2 ;

mergesort(a, first , mid);

mergesort(a,mid+1 , last);

merge(a,first,mid,last);

}

}

void merge(int a[] , int first , int mid , int last)

{

int b[50];

int i , j , k ;

i = first ;

j = mid + 1 ;

k = first ;

while(i <= mid && j<= last)

{

if(a[i]<= a[j])

b[k++]=a[i++];

else

b[k++]=a[j++];

}

if(i>mid)

{

while(j<=last)

b[k++] = a[j++];

}

else

{

while(i<=mid)

b[k++] = a[i++];

}

for(i=first;i<=last;i++)

a[i] = b[i];

}

void quicksort(int a[40] , int first , int last)

{

int i , j , pivot , temp ;

if(first < last)

{

pivot = first ;

i = first ;

j = last ;

while (i < j)

{

while (a[i] <= a[pivot] && i < last)

i++ ;

while(a[j] > a[pivot])

j--;

if(i<j)

{

temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

temp = a[pivot] ;

a[pivot] = a[j];

a[j] = temp;

quicksort(a,0,j-1);

quicksort(a,j+1,last);

}

}