**// Merge sort, tested on https://www.onlinegdb.com/online\_c\_compiler**

#include<stdio.h>

#include<conio.h>

int a[50];

void merge(int, int, int);

void merge\_sort(int low, int high);

{

int mid;

if(low<high)

{

mid=(low+high)/2;

merge\_sort(low, mid);

merge\_sort(mid+1, high);

merge(low, mid, high);

}

}

void merge(int low, int mid, int high)

{

int h, i, j, b[50], k;

h=low;

i=low;

j=mid+1;

while((h<=mid)&&(j<=high))

{

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

{

b[i]=a[h];

h++;

}

else

{

b[i]=a[j];

j++;

}

i++;

}

if(h>mid)

{

for(k=j;k<=high;k++)

{

b[i]=a[k];

i++;

}

}

else

{

for(k=h;k<=mid;k++)

{

b[i]=a[k];

i++;

}

}

for(k=low;k<=high;k++) a[k]=b[k];

}

int main()

{

int num,i;

printf("\n Enter the total numbers: ");

scanf("%d", &num);

printf("\n Enter %d numbers: \n", num);

for(i=1;i<=num; i++)

{

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

}

merge\_sort(1, num);

printf("\n SORTED ORDER: \n");

for(i=1; i<=num;i++)

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

getch();

}

**// binary search, tested on https://www.onlinegdb.com/online\_c\_compiler**

#include<stdio.h>

#include<conio.h>

int a[100], search;

int bsearch(int,int);

int main()

{

int c, n,loc;

printf("Enter number of elements: ");

scanf("%d", &n);

printf("\nEnter %d integers in sorted sorted :\n", n);

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

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

printf("\n Enter value to be searched in above list : ");

scanf("%d", &search);

loc=bsearch(1, n);

if(loc==-1)

printf("\n %d is not in the array\n", search);

else

printf("\n %d is available in the array at %d location\n", search,loc+1);

getch();

return 0;

}

int bsearch (int low, int high)

{

int mid;

if(low>high)

return -1;

else

{

mid=(low+high)/2;

if(a[mid]==search)

return mid;

else

{

if(search<a[mid])

bsearch(low,mid-1);

else

bsearch(mid+1,high);

}

}

}

**// quick sort, tested on https://www.onlinegdb.com/online\_c\_compiler**

#include<stdio.h>

#include<conio.h>

int a[40];

void quicksort(int p,int r);

int partition(int p,int r);

void exchange(int i,int j);

void quicksort(int p,int r)

{

int q;

if(p<r)

{

q=partition(p,r+1);

quicksort(p,q-1);

quicksort(q+1,r);

}

}

int partition(int p,int q)

{

int v,i,j;

v=a[p];

i=p;

j=q;

do

{

do

{

i++;

}while(a[i]<v);

do

{

j--;

}while(a[j]>v);

if(i<j)

{

exchange(i,j);

}

}while(i<j);

a[p]=a[j];

a[j]=v;

return j;

}

void exchange(int i,int j)

{

int temp;

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

int main()

{

int n,i;

printf("\nEnter no elements needed :" );

scanf("%d",&n);

printf("\nEnter elements :");

for(i=1;i<=n;i++)

{

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

}

a[n+1]=3276;

quicksort(1,n);

printf("\nSorted Array is :");

for(i=1;i<=n;i++)

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

getch();

return(0);

}