#include<stdio.h>

void Lsearch(int B[], int n);

void Bsearch(int B[], int n);

int main() {

int A[10], N;

printf("enter the number of elements: ");

scanf("%d", &N);

printf("enter the elements:\n");

for(int i = 0; i < N; i++) {

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

}

printf("1:Linear search\n2:Binary Search\n");

int choice;

printf("Choose 1 or 2: ");

scanf("%d", &choice);

switch(choice) {

case 1:

Lsearch(A, N);

break;

case 2:

Bsearch(A, N);

break;

default:

printf("wrong choice, choose 1 or 2\n");

break;

}

return 0;}

void Lsearch(int B[], int n) {

int val, i, found = 0;

printf("enter the value you want to search: ");

scanf("%d", &val);

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

if(B[i] == val) {

printf("%d is at location %d\n", val, i + 1);

found = 1;

}

}

if(found == 0) {

printf("value not found.\n");

}

}

void Bsearch(int B[], int n) {

int val;

for(int i = 0; i < n - 1; i++) {

for(int j = i + 1; j < n; j++) {

if(B[i] > B[j]) {

int temp = B[i];

B[i] = B[j];

B[j] = temp;

}

}}

printf("sorted array:\n");

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

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

}printf("\n");

printf("enter the value you want to search: ");

scanf("%d", &val);

int beg = 0, end = n - 1, mid, found = 0;

while(beg <= end) {

mid = (beg + end) / 2;

if(B[mid] == val) {

printf("%d is at location %d\n", val, mid + 1);

found = 1;

break;

}

else if(val > B[mid]) {

beg = mid + 1;

}

else if(val < B[mid]) {

end = mid - 1;

}}

if(found == 0) {

printf("value not found.\n");

}

}





