**TASK # 1:**

Write & implement the algorithm of Linear Search.

ALGORITHM:

**Input** = I, A, X, N.

**Output** = Found the required item.

**Step # 1:**(Initialization)

I, A, X, N.

**Step # 2:**

Read I, A, X, N.

**Step # 3:**

FOR(I=0;I<N;I++)

IF(A[I]==X) [Item Found]

[End of for loop]

IF(I>N)

[Element not found]

**Step # 4:**

Write X.

**Step # 5:**

Exit.

SOURCE CODE:

import java.util.Scanner;

public class LinearSearch {

public static void main(String[] args) {

Scanner inp=new Scanner(System.in);

int i,x;

int a[]=new int[5];

System.out.println("Enter elements");

for(i=0;i<5;i++){

a[i]=inp.nextInt(); }

System.out.println("Search Element");

x=inp.nextInt();

for(i=0;i<5;i++){

if(a[i]==x){

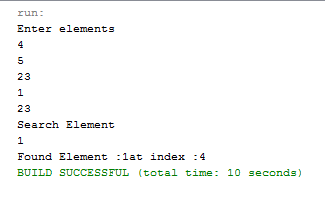
System.out.println("Found Element :"+x+"at index :"+(i+1));

Break; } }

if(i==5){

System.out.println("Not found");}}}

OUTPUT:



**TASK # 2:**

Write & implement the algorithm of Binary Search.

ALGORITHM:

**Input** = MID, LOW, HIGH, V, I, A, N.

**Output** = Find the location of derived output.

**Step # 1:**(Initialization)

MID, LOW=0, HIGH=N-1, V, I, A.

**Step # 2:**

Read MID, LOW=0, HIGH=N-1, V, I, A.

**Step # 3:**

MID=LOW+HIGH/2

WHILE(LOW<=HIGH)

IF(A[MID]>V)

LOW=MID+1

ELSE IF(A[MID]==V)

[Found Element]

ELSE

HIGH=MID-1

[End of while loop]

IF(LOW>HIGH)

[Element not found]

**Step # 4:**

Write V.

**Step # 5:**

Exit.

SOURCE CODE:

import java.util.Arrays;

import java.util.Scanner;

public class binarysearch {

public static void main(String[] args) {

Scanner inp=new Scanner(System.in);

int a[]=new int[5];

int i,v,low,high,mid;

System.out.println("Enter elements");

for(i=0;i<5;i++){

a[i]=inp.nextInt(); }

Arrays.sort(a);

System.out.println("Sorted array"+Arrays.toString(a));

System.out.println("Search Element");

v=inp.nextInt();

low=0;

high=a.length-1;

mid=low+high/2;

while(low<=high){

if(a[mid]<v){

low=mid+1; }

else if(a[mid]==v){

System.out.println("found value :"+v+"at Index :"+(mid+1));

break; }

else{

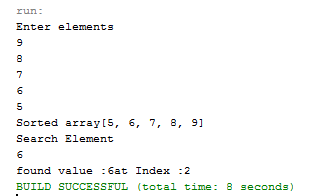
high=mid-1; }

mid=low+high/2; }

if(low>high){

System.out.println("Value not found"); } }}

OUTPUT:



**TASK # 3:**

Write & implement the algorithm of Matrix Multiplication.

ALGORITHM:

**Input** = A, B, C, I, J, K, N.

**Output** = Multiplication of Matrix.

**Step # 1:**(Initialization)

A, B, C, I, J, K, N.

**Step # 2:**

Read A, B, C, I, J, K, N.

**Step # 3:**

FOR(I=0;I<N;I++)

FOR(J=0;J<N;j++)

FOR(K=0;K<N;K++)

C[I][J]=C[I][J]+A[I][K]+B[K][J]

[End of Loops]

**Step # 4:**

Write C.

**Step # 5:**

Exit.

SOURCE CODE:

import java.util.Arrays;

public class matrixmult {

public static void main(String[] args) {

int a[][]={{1,2},{3,4}};

int b[][]={{5,6},{7,8}};

int c[][]=new int[a.length][b.length];

int i,j,k;

for(i=0;i<a.length;i++){

for(j=0;j<b.length;j++){

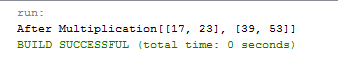
for(k=0;k<c.length;k++){

c[i][j]=c[i][j]+a[i][k]\*b[j][k];

}} }

System.out.println("After Multiplication"+Arrays.deepToString(c)); }}

OUTPUT:



**TASK # 4:**

Implement the binary search to add, remove, an element using Array class(char, int, double).

SOURCE CODE:

import java.util.Arrays;

public class BNsearch {

public static void main(String[] args) {

int a[]={45,3,12,2,4,5};

char b[]={'d','c','b','a'};

float c[]={5.2f,34.3f,2.3f};

int d[]={45,3,12,2,4,5};

int e[]=new int[6];

int f[]={9,8,7};

Arrays.sort(a);

System.out.println(Arrays.toString(a));

System.out.println(Arrays.binarySearch(a,45));

Arrays.sort(b);

System.out.println(Arrays.toString(b));

System.out.println(Arrays.binarySearch(b,'c'));

Arrays.sort(c);

System.out.println(Arrays.toString(c));

System.out.println(Arrays.binarySearch(c,5.2f));

System.out.println(Arrays.equals(a,d));

int[] a2=Arrays.copyOf(f,5);

a2[3]=8;

a2[4]=9;

System.out.println(Arrays.toString(a2));

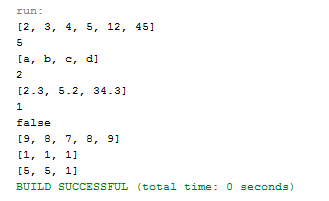
Arrays.fill(f,1);

System.out.println(Arrays.toString(f));

Arrays.fill(f,0,2,5);

System.out.println(Arrays.toString(f)); }}

OUTPUT:

****