OUT PARAMETERS

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Day20

{

class Program

{

static public bool Rectangle(int l,int b,out int area,out int peri)

{

area = l \* b;

peri = 2 \* (l + b);

return true;

}

static void Main(string[] args)

{

int len = 20, bred = 2, a, p;

bool res = Rectangle(len, bred, out a, out p);

Console.WriteLine("LENGTH = " + len);

Console.WriteLine("BREADTH = " + bred);

Console.WriteLine("AREA = " + a);

Console.WriteLine("PERIMETER = " + p);

Console.WriteLine("RES = " +res);

}

}

}

INSERTION SORT

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Day20

{

class Insertionsort

{

static void insertion(int[] a)

{

int i, j, key,n;

n = a.Length;

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

{

key = a[i];

j = i - 1;

while (j >= 0 && a[j] > key)

{

a[j + 1] = a[j];

j--;

}

a[j + 1] = key;

}

}

public static void Main(string[] args)

{

int n;

Console.WriteLine("Enter the n value ");

n = Convert.ToInt32(Console.ReadLine());

int[] a = new int[n];

Console.WriteLine("enter the array elements ");

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

a[i] = Convert.ToInt32(Console.ReadLine());

insertion(a);

Console.WriteLine("The sorted array elements ");

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

Console.WriteLine(a[i]);

}

}

}

SELECTION SORT

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Day20

{

class SelectionSort

{

static void selection(int[] a)

{

int i,j, minind, n,temp;

n = a.Length;

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

{

minind = i;

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

{

if (a[j] < a[minind])

minind = j;

}

temp = a[i];

a[i] = a[minind];

a[minind] = temp;

}

}

public static void Main(string[] args)

{

int n;

Console.WriteLine("Enter the n value ");

n = Convert.ToInt32(Console.ReadLine());

int[] a = new int[n];

Console.WriteLine("enter the array elements ");

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

a[i] = Convert.ToInt32(Console.ReadLine());

selection(a);

Console.WriteLine("The sorted array elements ");

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

Console.WriteLine(a[i]);

}

}

}

LINEAR SEARCH

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Day20

{

class LinearSearch

{

static int linear(int[] a,int key)

{

int i=0,n=a.Length;

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

{

if (key == a[i])

return i;

}

return -1;

}

public static void Main(string[] args)

{

int n,key;

Console.WriteLine("Enter the n value ");

n = Convert.ToInt32(Console.ReadLine());

int[] a = new int[n];

Console.WriteLine("enter the array elements ");

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

a[i] = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter the key value ");

key = Convert.ToInt32(Console.ReadLine());

int res = linear(a,key);

if(res==-1)

Console.WriteLine("Did not find the key");

else

Console.WriteLine("Found key = {0} at index= {1}",key,res);

}

}

}

BINARY SEARCH

**SCENARIO 1 (RIGHT SIDE)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | |  | |  | |  | |  | |
|  | 11 | 22 | 33 | 44 | 55 | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | | if(a[mid]==key) | | | |
|  |  |  |  |  | (low+high)/2 | |  | |  | | Found at | | mid | |
| n= 5 |  |  | low | high | mid | | a[mid] | |  | |  | |  | |
| low | 0 |  | 0 | 4 | 2 | | 33 | |  | | if(a[mid]<key) | | | |
| high | n-1=4 |  | 3 | 4 | 3 | | 44 | |  | | low=mid+1 | | | |
| key | 55 |  | 4 | 4 | 4 | | 55 | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |
|  |  |  |  |  |  | |  | |  | |  | |  | |
|  | 0 | 1 | 2 | 3 | 4 | |  | |  | |  | |  | |
|  | 11 | 22 | 33 | 44 | 55 | |  | |  | |  | |  | |
|  |  |  |  |  |  |  | |  | | if(a[mid]==key) | | | |
| **SCENARIO 2 (LEFT SIDE)** |  |  |  |  | (low+high)/2 |  | |  | | Found at | | mid | |
| n= 5 |  |  | low | high | mid | a[mid] | |  | |  | |  | |
| low | 0 |  | 0 | 4 | 2 | 33 | |  | | if(a[mid]<key) | | | |
| high | n-1=4 |  | 0 | 1 | 0 | 11 | |  | | low=mid+1 | | | |
| key | 11 |  |  |  |  |  | |  | |  | |  | |
|  |  |  |  |  |  |  | |  | | IF(A[MID]>KEY) | | | |
|  |  |  |  |  |  |  | |  | | HIGH=MID-1 | | | |
|  |  |  |  |  |  |  | |  | |  | |  | |
|  |  |  |  |  |  |  | |  | |  | |  | |

SCENARIO 3: 10

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 |  |  |  |  |
|  | 11 | 22 | 33 | 44 | 55 |  |  |  |  |
|  |  |  |  |  |  |  |  | if(a[mid]==key) | |
|  |  |  |  |  | (low+high)/2 |  |  | Found at | mid |
| n= 5 |  |  | low | high | mid | a[mid] |  |  |  |
| low | 0 |  | 0 | 4 | 2 | 33 |  | if(a[mid]<key) | |
| high | n-1=4 |  | 0 | 1 | 0 | 11 |  | low=mid+1 | |
| key | 10 |  | 0 | -1 |  |  |  |  |  |
|  |  |  |  |  |  |  |  | IF(A[MID]>KEY) | |
|  |  |  |  |  |  |  |  | HIGH=MID-1 | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

SCENARIO 4: 66

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 |  |  |  |  |
|  | 11 | 22 | 33 | 44 | 55 |  |  |  |  |
|  |  |  |  |  |  |  |  | if(a[mid]==key) | |
|  |  |  |  |  | (low+high)/2 |  |  | Found at | mid |
| n= 5 |  |  | low | high | mid | a[mid] |  |  |  |
| low | 0 |  | 0 | 4 | 2 | 33 |  | if(a[mid]<key) | |
| high | n-1=4 |  | 3 | 4 | 3 | 44 |  | low=mid+1 | |
| key | 66 |  | 4 | 4 | 4 | 55 |  |  |  |
|  |  |  | 5 | 4 |  |  |  | IF(A[MID]>KEY) | |
|  |  |  |  |  |  |  |  | HIGH=MID-1 | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

SCENARIO 5:22

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 |  |  |  |  |
|  | 11 | 22 | 33 | 44 | 55 |  |  |  |  |
|  |  |  |  |  |  |  |  | if(a[mid]==key) | |
|  |  |  |  |  | (low+high)/2 |  |  | Found at | mid |
| n= 5 |  |  | low | high | mid | a[mid] |  |  |  |
| low | 0 |  | 0 | 4 | 2 | 33 |  | if(a[mid]<key) | |
| high | n-1=4 |  | 0 | 1 | 0 | 11 |  | low=mid+1 | |
| key | 22 |  | 1 | 1 | 1 | 22 |  |  |  |
|  |  |  |  |  |  |  |  | IF(A[MID]>KEY) | |
|  |  |  |  |  |  |  |  | HIGH=MID-1 | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Day20

{

class BinarySearch

{

static int binary(int[] a,int key)

{

int low, high, mid, n;

n = a.Length;

low = 0;

high = n - 1;

while(low<=high)

{

mid = (low + high) / 2;

if (a[mid] == key)

return mid;

else if (a[mid] < key)

low = mid + 1;

else

high = mid - 1;

}

return -1;

}

public static void Main(string[] args)

{

int n, key;

Console.WriteLine("Enter the n value ");

n = Convert.ToInt32(Console.ReadLine());

int[] a = new int[n];

Console.WriteLine("enter the array elements in sorted order ");

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

a[i] = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter the key value ");

key = Convert.ToInt32(Console.ReadLine());

int res = binary(a, key);

if(res==-1)

Console.WriteLine("Search key not found");

else

Console.WriteLine("search key={0} found at ={1}",key,res);

}

}

}

DOUBLE DIMENSIONAL ARRAY

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Day20

{

class Program

{

public static void Main(string[] args)

{

//hardcoding the array elements

int[,] a = { { 22, 3, 4 }, { 33, 4, 5 } };

// Console.WriteLine("rows = " + a.GetLength(0));

// Console.WriteLine("cols = " + a.GetLength(1));

int rows = a.GetLength(0);

int cols = a.GetLength(1);

for (int i = 0; i < 2; i++)

{

for (int j = 0; j < 3; j++)

{

Console.Write(a[i, j] + "\t");

}

Console.WriteLine();

}

}

}

**2D ARRAY USER INPUT**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Day20

{

class Program

{

public static void Main(string[] args)

{

int rows, cols;

Console.WriteLine("Enter the no of rows = ");

rows = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter the no of columns = ");

cols = Convert.ToInt32(Console.ReadLine());

int[,] a = new int[rows,cols];

Console.WriteLine("Enter the array elements ");

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

{

for (int j = 0; j < cols; j++)

{

a[i, j] = Convert.ToInt32(Console.ReadLine());

}

}

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

{

for (int j = 0; j < cols; j++)

{

Console.Write(a[i, j] + "\t");

}

Console.WriteLine();

}

}

}

}