2. Write a program to print if the given day is a working day or holiday. Accept a number representing the day from the user and apply below logic

1. The number should be > 0 and <=7.
2. If the number is b/n 1 to 5 it is a working day
3. If the number is >5 – holiday

namespace project1

{

class workingorholiday

{

static void Main()

{

int dayno = int.Parse(Console.ReadLine());

switch(dayno)

{

case 1:

case 2:

case 3:

case 4:

case 5:

Console.WriteLine($"Working day");

break;

case 6:

case 7:

Console.WriteLine($"Holiday");

break;

default:

Console.WriteLine($"enter valid day number");

break;

}

}

}

}

8. Modify q2 to so that it returns both the position and also the number

using System;

namespace weekday

{

class Program

{

enum week { Monday=1,Tuesday,Wednesday,Thursday,Friday,Saturday,Sunday};

static void Main(string[] args)

{

week mon = week.Monday;

week tue = week.Tuesday;

week wed = week.Wednesday;

week thu = week.Thursday;

week fri = week.Friday;

week sat = week.Saturday;

week sun = week.Sunday;

int mon1 = (int)week.Monday;

int tue1 = (int)week.Tuesday;

int wed1 = (int)week.Wednesday;

int thu1 = (int)week.Thursday;

int fri1 = (int)week.Friday;

int sat1 = (int)week.Saturday;

int sun1 = (int)week.Sunday;

Console.WriteLine("Enter a number between 1 and 7");

String n = Console.ReadLine();

int num = int.Parse(n);

if (num > 0 && num <= 7)

{

switch (num)

{

case 1:

Console.WriteLine(mon + " is a working day");

Console.WriteLine("Position "+mon1);

break;

case 2:

Console.WriteLine(tue + " is a working day");

Console.WriteLine("Position " + tue1);

break;

case 3:

Console.WriteLine(wed + " is a working day");

Console.WriteLine("Position " + wed1);

break;

case 4:

Console.WriteLine(thu + " is a working day");

Console.WriteLine("Position " + thu1);

break;

case 5:

Console.WriteLine(fri + " is a working day");

Console.WriteLine("Position " + fri1);

break;

case 6:

Console.WriteLine(sat + "is a holiday");

Console.WriteLine("Position " + sat1);

break;

case 7:

Console.WriteLine(sun + " is a holiday");

Console.WriteLine("Position " + sun1);

break;

}

}

else

{

Console.WriteLine("Out of range!");

}

}

}

1. Binary Search Algorithm

using System;

class GFG {

// Returns index of x if it is present in

// arr[l..r], else return -1

static int binarySearch(int[] arr, int l,

int r, int x)

{

if (r >= l) {

int mid = l + (r - l) / 2;

// If the element is present at the

// middle itself

if (arr[mid] == x)

return mid;

// If element is smaller than mid, then

// it can only be present in left subarray

if (arr[mid] > x)

return binarySearch(arr, l, mid - 1, x);

// Else the element can only be present

// in right subarray

return binarySearch(arr, mid + 1, r, x);

}

// We reach here when element is not present

// in array

return -1;

}

// Driver method to test above

public static void Main()

{

int[] arr = { 2, 3, 4, 10, 40 };

int n = arr.Length;

int x = 10;

int result = binarySearch(arr, 0, n - 1, x);

if (result == -1)

Console.WriteLine("Element not present");

else

Console.WriteLine("Element found at index "

+ result);

}

}

1. Bubble sort

using System;

namespace BubbleSort

{

class MySort

{

static void Main(string[] args)

{

int[] arr = { 71, 15, 95, 33, 72, 18 };

int temp;

for (int j = 0; j <= arr.Length - 2; j++)

{

for (int i = 0; i <= arr.Length - 2; i++)

{

if (arr[i] > arr[i + 1])

{

temp = arr[i + 1];

arr[i + 1] = arr[i];

arr[i] = temp;

}

}

}

Console.WriteLine("Sorted:");

foreach (int p in arr)

Console.Write(p + " ");

Console.Read();

}

}