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

Hint: use Enums

Solution:

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;

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");

break;

case 2:

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

break;

case 3:

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

break;

case 4:

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

break;

case 5:

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

break;

case 6:

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

break;

case 7:

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

break;

}

}

else

{

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

}

}

}

}

8. Modify Q2 to so that it returns both the position and also the number

Solution:

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!");

}

}

}

}

DS&A

1. Binary search algorithm
2. Bubble sort algorithm

1.Solution:

using System;

class Binarysearch

{

static int binarySearch(int[] arr, int l, int r, int x)

{

if (r >= l)

{

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

if (arr[mid] == x)

return mid;

if (arr[mid] > x)

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

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

}

return -1;

}

static void bubbleSort(int[] arr)

{

int n = arr.Length;

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

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

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

{

int temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

public static void Main()

{

int n;

Console.WriteLine("Enter length of the array");

String a = Console.ReadLine();

n = int.Parse(a);

int[] a1 = new int[n];

Console.WriteLine("Enter " + n + " elements");

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

{

String v = Console.ReadLine();

int v1 = int.Parse(v);

a1[i] = v1;

}

int len = a1.Length;

int x;

int min = 0;

int max = len - 1;

bubbleSort(a1);

Console.WriteLine("Enter the element to be searched");

String ele = Console.ReadLine();

x = int.Parse(ele);

int result = binarySearch(a1, min, max, x);

if (result == -1)

Console.WriteLine("Element not present");

else

Console.WriteLine("Element found at index "+ result);

}

}

Solution 2:

using System;

class Bubblesort

{

static void bubbleSort(int[] arr)

{

int n = arr.Length;

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

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

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

{

int temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

static void printArray(int[] arr)

{

int n = arr.Length;

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

Console.Write(arr[i] + " ");

Console.WriteLine();

}

public static void Main()

{

int n;

Console.WriteLine("Enter length of the array");

String a = Console.ReadLine();

n = int.Parse(a);

int[] a1 = new int[n];

Console.WriteLine("Enter " +n+ " elements");

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

{

String v = Console.ReadLine();

int v1 = int.Parse(v);

a1[i] = v1;

}

bubbleSort(a1);

Console.WriteLine("Sorted array");

printArray(a1);

}

}