**“La tecnología es solo una herramienta, nosotros somos lo más importante”**

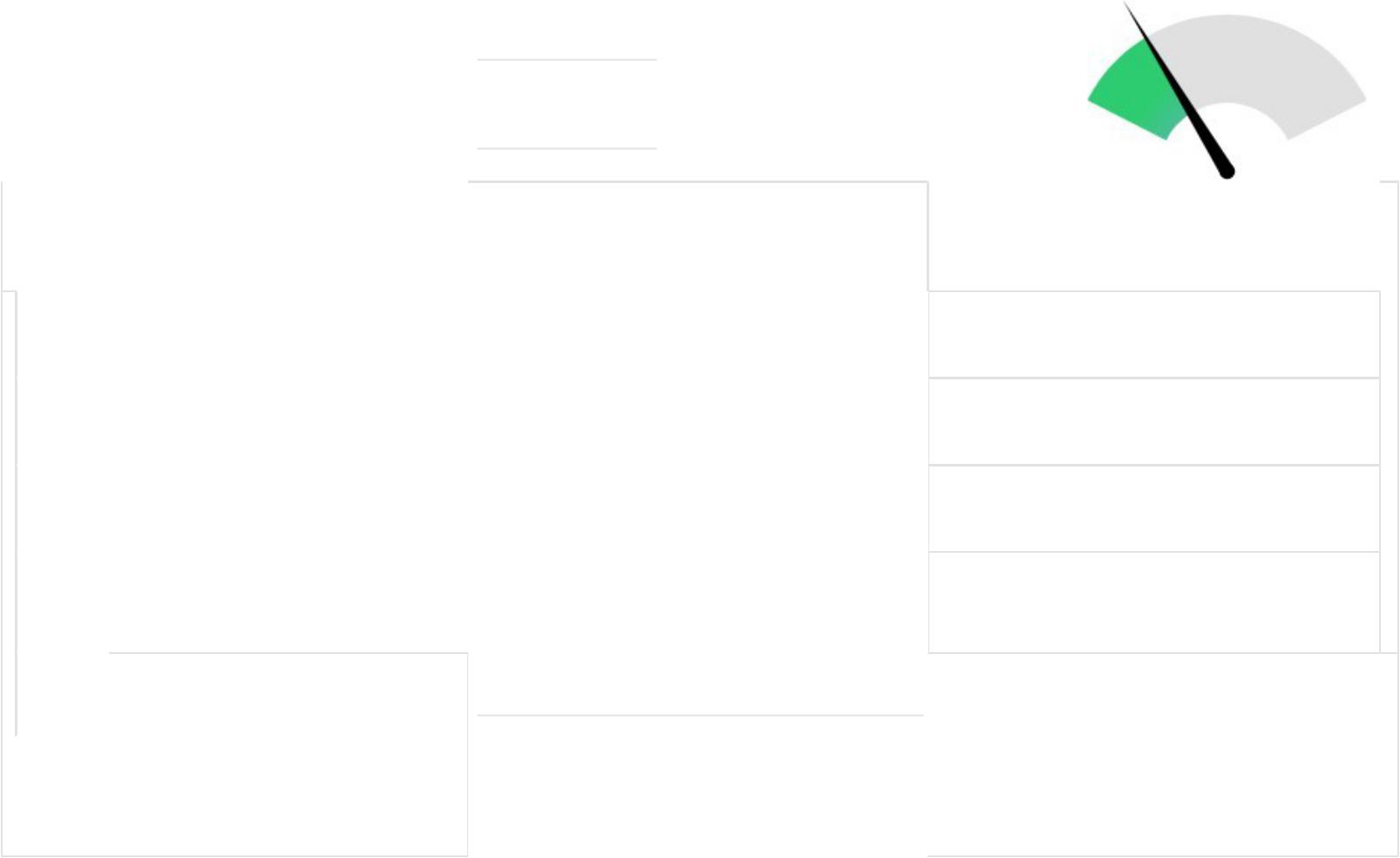
Prueba Básica C# (.Net)

Carlos Ramos

**Fecha Inicio:** 2019-02-22 12:37:35 **Fecha Fin:** 2019-02-22 13:45:32

**Resultado:**

El candidato demuestra tener buenas habilidades de programación por lo que puede resolver problemas de dificultad media e incluso alta.



**Tiempo Total:** 1 Hora 7 Minutos 57Segundos

**Puntaje Total:**

85/100

Habilidad en Programación Alto

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Información Candidato | | Información Prueba | | | | |  | Información Instancia | |  |  |
|  |  |  |  |  |  | |  |  |  |  |  |  |  |
|  |  | **Id: 3384** | | **Descripción: Prueba Básica C#** | | | | | | **Límite de Días:** | **0** |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | **Nombre: Carlos Ramos** | |  | **(.Net)** |  |  |  |  | **Tiempo Límite:** | **90 minutes** |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | |  |  |  |  |  |  |  |  |  |  |
|  |  | **Notas:** | | **Descripción Larga:** | |  |  |  |  | **Tipo: TST** |  |  |  |
|  |  |  |  | Prueba Básica C# (.Net) con preguntas de | | | | | |  |  |  |  |
|  |  | **Registrado: 2019-02-22** | | **Proceso: .NET Visual Basic** | |  |  |
|  |  | dificultad menor a media y con baja exigencia | | | | | |  |  |
|  |  |  | **carlosesteban943@gmail.** | de tiempo. | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | **Email: com** | | **Nivel de Dificultad:** | | **Medio** | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |

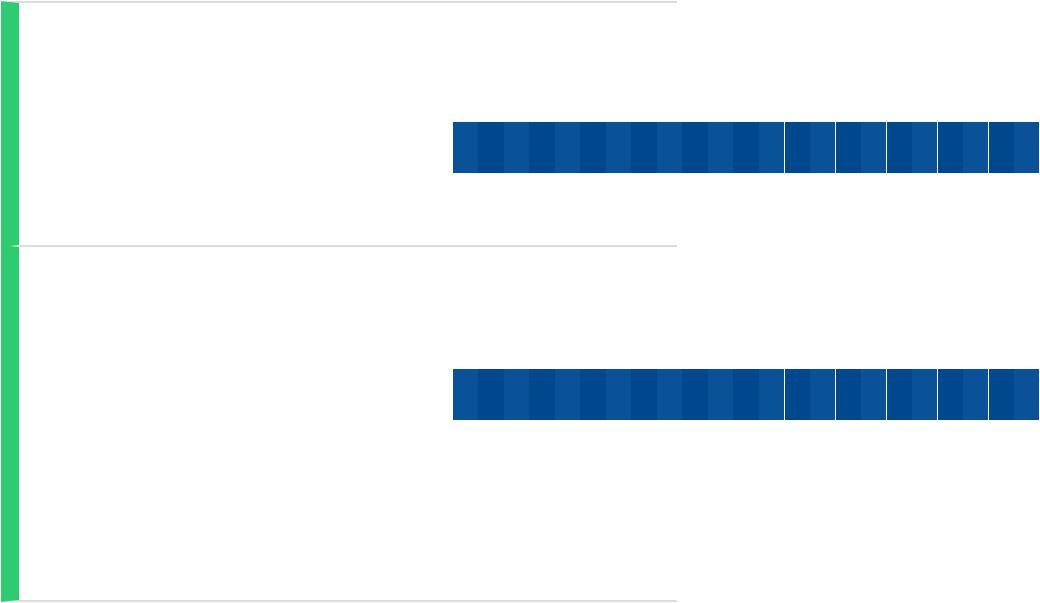
**Resumen por Categoría**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Descripción** | **Resultado** | **Puntaje (Valor)** | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Programación |  |  |  |  |  |  |  |  | 8 | 5/ | 1 | 0 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  | 0 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |



**Resultado Categoría:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  | Puntaje > 66% | |  |
|  |  |  |  |  |  |  | 33% < Puntaje < 66% | |  |
|  |  |  |  |  |  |  | Puntaje < 33% | |  |
|  | **Resumen por Pregunta** | | |  |  |  |  |  |  |
|  |  |  |  |  |  | |  |  |  |
|  | **Descripción** |  |  | **Categoria** | **Nivel de Dificultad Puntaje** | | | |  |
|  |  | |  |  |  |  |  |  |  |
|  | 1.- Encuentra el mayor número en un arreglo | | | PROG | Fácil |  | 20 de 20 | |  |
|  | **Escenarios de Validación:** |  |  | 100 % |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  | 2.- Crear Histograma en base a arreglo |  |  | PROG | Medio |  | 28 de 40 | |  |
|  | **Escenarios de Validación:** |  |  | 100 % |  |  |  |  |  |
|  | **Eficiencia del Código:** |  |  | 0% |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  | |  | |  |  |  |  |  |
|  | 3.- Encuentra el numero con la secuencia mas larga en el PROG | | | | Medio |  | 37 de 40 | |  |
|  | arreglo |  |  |  |  |  |  |  |  |
|  | **Escenarios de Validación:** |  |  | 100 % |  |  |  |  |  |
|  | **Eficiencia del Código:** | 68 % | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |



 No Respondido



Parcialmente Correcto

 Incorrecto



Correcto

**Detalles por Pregunta**



1.- Encuentra el mayor número en un arreglo

**Tipo:** Programming Question with predefined code

**Categoria:** PROG

**Puntaje:** 20 of 20

**Nivel de Dificultad:** Fácil

**Familia:** Programación

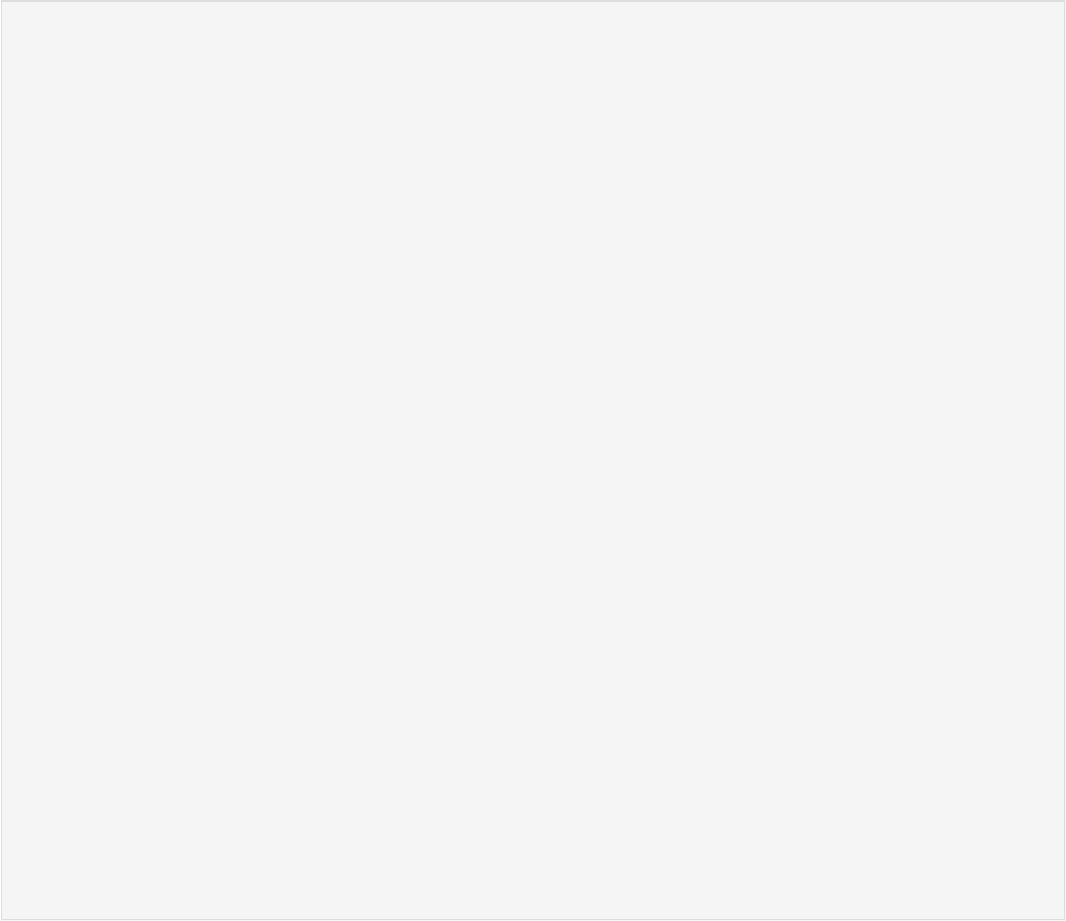
**SubFamilia:** C# (.Net) **Texto**

**de la Pregunta:**

Tienes un arreglo (llamado myArray) con 5 elementos (enteros en el rango de 1 a 100). Escribe un programa en C# que imprima el numero mas alto del arreglo (Si se repite, solo imprimir una vez). El programa solo debe imprimir el número, sin ningún texto.

El código que llena el arreglo ya esta escrito, pero puedes editarlo para probar con otros valores. Con el botón de refrescar puedes recuperar el valor original que sera utilizado para evaluar la pregunta como correcta o incorrecta durante la ejecución.

**Respuesta:**



int mayor = 0;

for (int i = 0; i < myArray.Length; i++){ if(myArray[i] < 1 || myArray[i]>100 ){ Console.WriteLine("enteros fuera del rango de 1 a 100");

break;

}

if(i > mayor){ mayor=

myArray[i];

}

}

if(mayor > 0)

{ Console.WriteLine(mayor

);

Escenarios de Validación



|  |  |
| --- | --- |
| **Descripción** | **Estado** |
|  |  |
| Caso Básico | Correcto |
|  |  |
| Numero Doble | Correcto |

Caso ignorar formato Correcto

2.- Crear Histograma en base a arreglo

**Tipo:** Programming Question with predefined code

**Categoria:** PROG

**Puntaje:** 28 of 40

**Nivel de Dificultad:** Medio

**Familia:** Programación

**SubFamilia:** C# (.Net) **Texto**

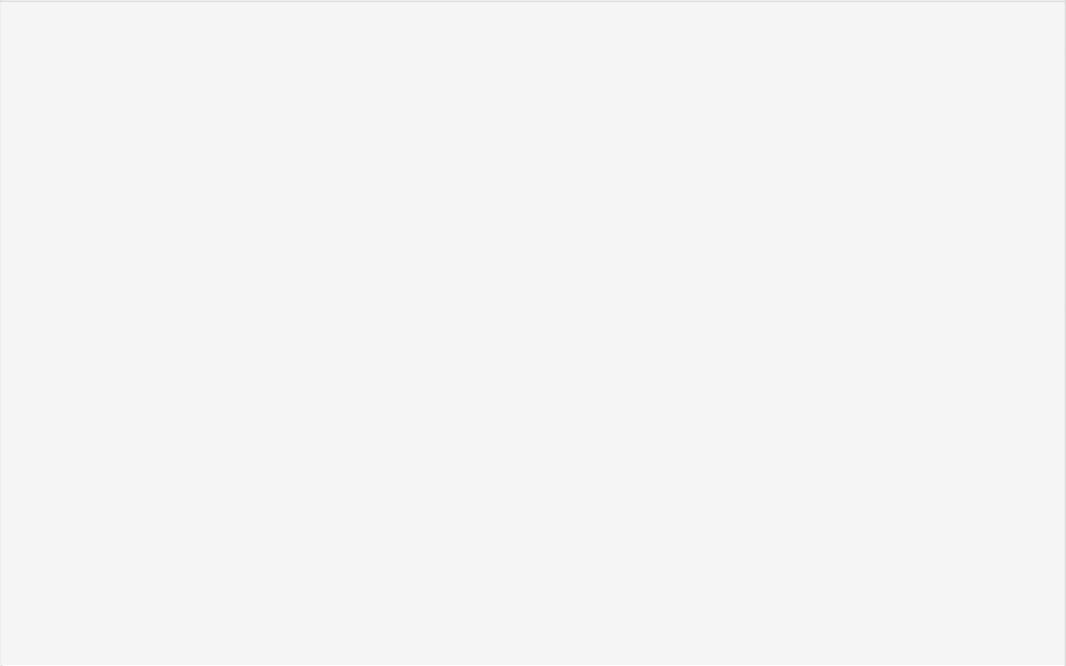
**de la Pregunta:**

Escribir un programa en C# que recorra un arreglo y genere un histograma en base a los números de este. El arreglo se llama myArray y contiene 10 elementos que corresponden a números enteros del 1 al 5. Un histograma representa que tanto un elemento aparece en un conjunto de datos. Por ejemplo, para el arreglo: myArray:=(1,2,1,3,3,1,2,1,5,1) el histograma se vería asi:

1. \*\*\*\*\*
2. \*\*
3. \*\*
4. \*

El código para declarar y poblar myArray ya está ahí, puede editarlo para probar con otros valores y puede hacer clic en el botón de actualizar junto a él para volver al valor original que se utilizará para validar su código durante la prueba. Asegúrese de que los resultados se impriman exactamente como aparecen aquí, ya que incluso un espacio faltante o sobrante puede marcar la pregunta como incorrecta (Notar espacio entre los ":" y el primer asterisco).

**Respuesta:**



string uno = "", dos = "", tres = "", cuatro = "", cinco = ""; bool valido = true;

for (int i = 0; i < myArray.Length; i++)

{ if(myArray[i] < 1 || myArray[i]>5 ) { valido = false;

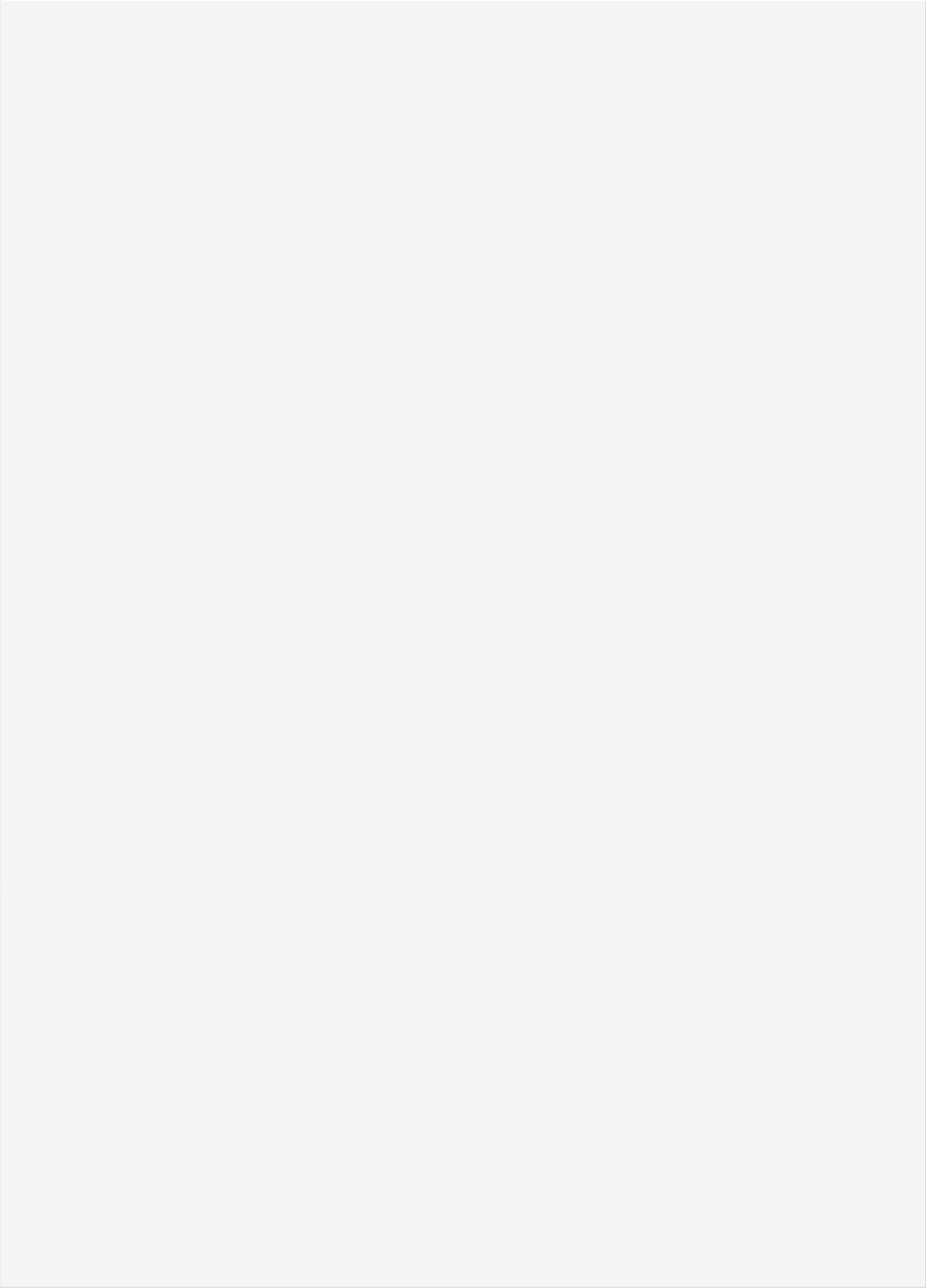
break;

}

switch(myArray[i])

{ case 1: uno += "\*";

break;

case 2: dos

+= "\*";

break; case

1. tres += "\*"; break; case 4: cuatro += "\*"; break;

case 5: cinco += "\*"; break;

}

}

if(!valido){

Console.WriteLine("enteros fuera del rango de 1 a 5");

}else{

Console.WriteLine("1: "+uno);

Console.WriteLine("2: "+dos);

Console.WriteLine("3: "+tres);

Console.WriteLine("4: "+cuatro);

Console.WriteLine("5: "+cinco);

}

Escenarios de Validación



|  |  |
| --- | --- |
| **Descripción** | **Estado** |
|  |  |
| Caso Extremo ignora Espacio | Correcto |
|  |  |
| Caso Básico Extremo | Correcto |
|  |  |
| Caso Básico ignora Espacio | Correcto |
|  |  |
| Caso Respuesta Aproximada | Correcto |

3.-

Encuentr

* el numero con la secuenci
* mas larga en el arreglo

**Tipo:** Programming

Question with

predefined code

**Categoria:** PROG

**Puntaje:** 37 of 40

**Nivel de Dificultad:**

Medio

**Familia:** Programación

**SubFamilia:** C# (.Net)

**Texto de la Pregunta:**

Tienes un arreglo (llamado myArray) con 10 elementos (enteros en el rango de 1 a 9). Escribe un programa que imprima el numero que tiene mas ocurrencias seguidas en arreglo y también imprimir la cantidad de veces que aparece en la secuencia.

El código que llena el arreglo ya esta escrito, pero puedes editarlo para probar con otros valores. Con el botón de refrescar puedes recuperar el valor original que sera utilizado para evaluar la pregunta como correcta o incorrecta durante la ejecución.

Su programa escrito en C# debe analizar el arreglo de izquierda a derecha

para que en caso de que dos números cumplan la condición, el que aparece

por primera vez de izquierda a derecha será el que se imprimirá. La salida

de los datos para el arreglo en el ejemplo (1,2,2,5,4,6,7,8,8,8) sería la

siguiente:

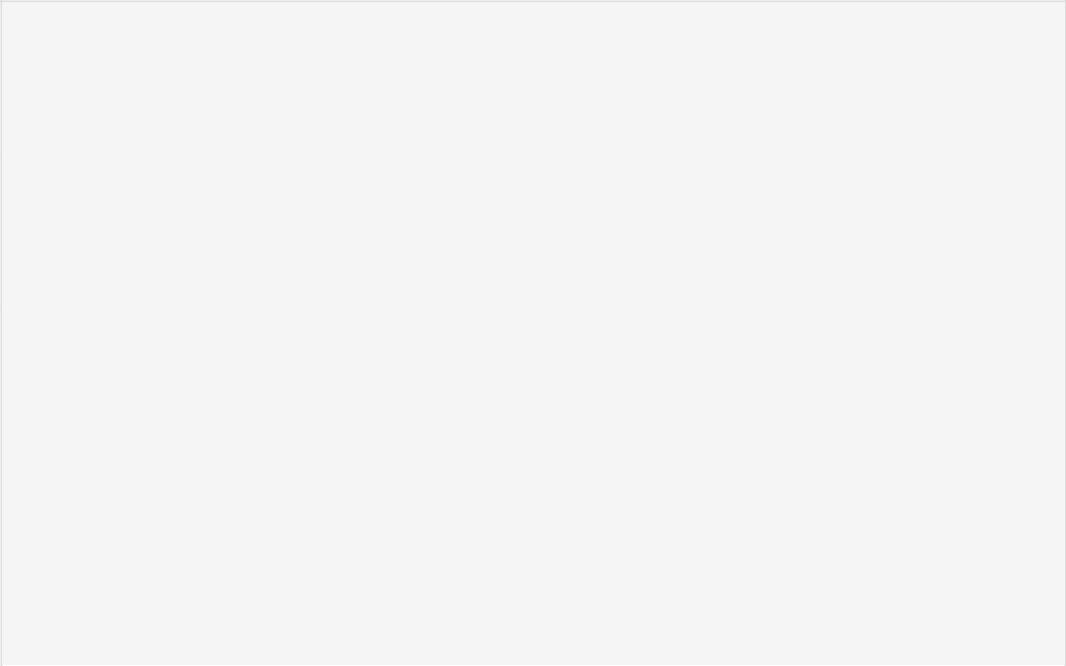
Longest: 3

Number: 8

En el ejemplo, la secuencia mas larga la tiene el numero 8 con una secuencia de tres ochos seguidos. Tenga en cuenta que el código que escriba debe imprimir los resultados exactamente como se muestra con el fin de que la pregunta sea considerada válida.

El código para declarar y poblar myArray ya está ahí, puede editarlo para probar con otros valores y puede hacer clic en el botón de actualizar junto a él para volver al valor original que se utilizará para validar su código durante la prueba.

**Respuesta:**



int mayorSec = 1;

int mayorSecAux = 1;

int numeroMayorSec = 0;

int numeroAux = 0;

for (int i = 0; i < myArray.Length; i++)

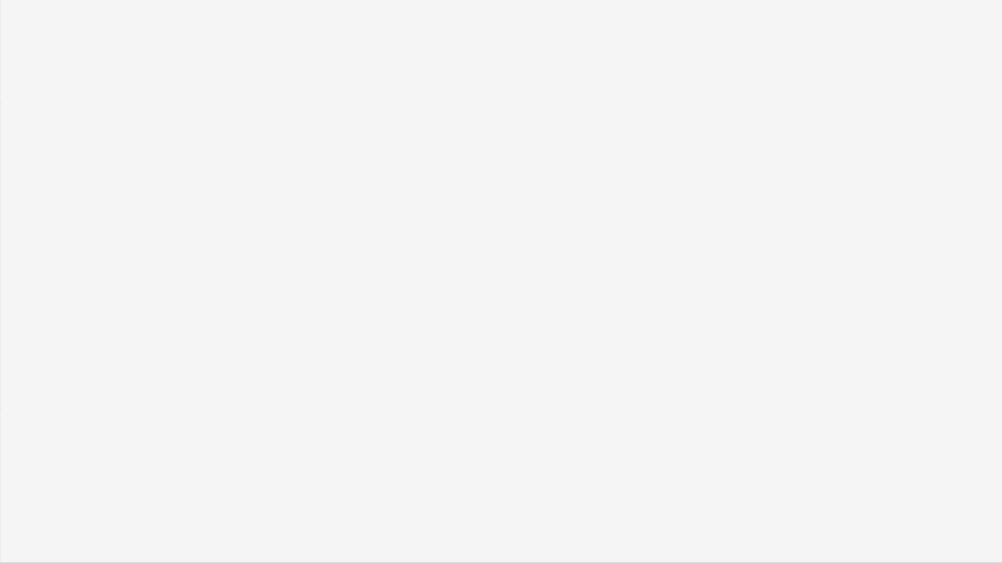
{ if(myArray[i] == numeroAux){ mayorSecAux +=1;

}else{ mayorSecAux

=0;

}

**“La tecnología es solo una herramienta, nosotros somos lo más importante”**



if (mayorSecAux > mayorSec)

{ mayorSec = mayorSecAux;

numeroMayorSec = myArray[i];

}

numeroAux = myArray[i];

}

Console.WriteLine("Longest: "+(mayorSec + 1));

Console.WriteLine("Number: "+numeroMayorSec);

Escenarios de Validación



|  |  |  |
| --- | --- | --- |
| **Descripción** | | **Estado** |
|  |  |  |
| Todos iguales | | Correcto |
|  |  |  |
| Caso Básico Estricto | | Correcto |
|  |  |  |
| Caso doble | | Correcto |
|  |  |  |
| Caso Básico Simple | | Correcto |
|  |  |  |
| Secuencia Duplicada | | Correcto |
|  |  |  |