**ALGORITMOS 301303**

Nombre estudiante: Jorge Mario Rincón Gutiérrez

Grupo: 137

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Planteamiento del problema a resolver:  Un banco ha solicitado se diseñe un programa que permita encriptar la información de las contraseñas (4 números) digitada por teclado (como un único número) hasta el servidor principal, utilizando el siguiente criterio, el primer número se envía de último, el segundo, de penúltimo, el tercer número pasa a la segunda posición, el último pasa a ser primero: ejemplo: Sea 6532, se debe enviar como 2356 | | | | | | |
| Que datos no conozco y se deben solicitar (si se requiere más filas por favor anexarlas) | Que datos no conozco, pero los puedo calcular | Tipo  Variables requeridas | | Formulas a aplicar si se requieren | Requiere condicional (si / no) | Requiere ciclos (si / no) \_ cuales |
| Tipo | Variable |
| La clave solicitada al usuario | La clave codificada | int | clave |  |  |  |
|  |  | int | clavencriptada |  |  |  |
|  |  | int | num |  |  |  |
|  |  | int | respuesta |  | Si el número es de 4 dígitos |  |
|  |  | int | clavecodificada | Colocar numero Al revés dividiendo entre 10, obteniendo sus residuos y cociente al final |  | Si, dividir entre 10 hasta que no se pueda |
|  | Los residuos para voltear el numero | Int | nuevonum |  |  |  |
|  | Los residuos para voltear el numero | Int | residuo |  |  |  |

|  |
| --- |
| Restricciones a tener en cuenta – Filtros- |
| El numero debe de ser de 4 dígitos. |
|  |
|  |

**Código**:

//Autor: Jorge Mario Rincón G., Código 16539985.

//Fecha: 22-10-2017.

//Doy fe que este ejercicio es de mi autoría. En caso de encontrar plagio, la nota de todo mi trabajo debe ser de CERO; además, de las respectivas sanciones a que haya lugar.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Banco // encapsulación de las variables

{

class Program // nombre de la clase

{

static void Main(string[] args) // Método principal

{

mensajeInicial(); // mostrar el mensaje anti plagio

int clave= solicitarClave(); // solicitar la clave

if(validarClave(clave)==1) // si la clave es un numero de 4 digitos

encriptarClave(clave); // se encripta la clave

else

{

Console.Write("Clave Incorrecta,solo deben ser numeros y tener 4 digitos, no estar vacío"); // mensaje clave incorrecta

Console.ReadKey(); // esperar tecla para sostener pantalla

}

}

public static void mensajeInicial (){ // método publico para mostrar el mensaje de siempre

// mostrar el mensaje de siempre

Console.WriteLine("Autor: Jorge Mario Rincón, code 16539985");

Console.WriteLine("Fecha: 22-10-2017.");

Console.WriteLine("Doy fe que este ejercicio es de mi autoría. En caso de encontrar plagio, la nota de todo mi trabajo debe ser de CERO; además, de las respectivas sanciones a que haya lugar.");

Console.WriteLine("Encriptar la clave para el Banco, Presione tecla");

Console.WriteLine(""); //salto de linea

Console.ReadKey(); // sostener pantalla

Console.Clear(); // limpiar la consola

}

public static int solicitarClave() { // método para solicitar la clave

int num; // definir numero a solicitar

Console.WriteLine("Hola, Por favor digita la clave de 4 digitos"); // mostrar mensaje a usuario

num = int.Parse(Console.ReadLine()); // pedir el numero entero

return num; // devolver el numero solicitado por el usuario

}

public static int validarClave(int clave ) { // método para calidar la clave

int respuesta=0; // definir respuesta

if (clave > 0 && clave < 9999 ) // si la clave es de 4 digitos

respuesta = 1; // se envia uno validando correctamente

return respuesta; // de lo contrario se envia 0

}

public static void encriptarClave(int c=0) { // método para encriptar la clave con divisiones entre 10 ejemplo: 456/10 el residuo es 6 el cociente 45, 45/10 4 residuo 5, acumulando quedaria alrevés

string clavecodificada=""; // variable para guardar el dato encriptado

int nuevonum = c; // variable para guardar el nuvo numero mientras se divide entre 10 para obtener sus residuos acumulados y obtener el numero al réves

for (int residuo; nuevonum >= 10;){ // ciclo para dividir entre 10 hasta que no se pueda mas

residuo = nuevonum % 10; // se capturan los residuos

nuevonum = nuevonum / 10; // se capturan los cocientes

clavecodificada = clavecodificada + residuo; // se acumula el numero al revés

}

clavecodificada = clavecodificada + nuevonum; // se captura el ultimo numero cociente

Console.WriteLine("Clave Encriptada: "+clavecodificada); // se muestra la clave encriptada

Console.ReadKey(); // se sostiene pantalla

}

}

}

**Usted probo el programa con diferentes valores Si / No**? Si

(Cuales):

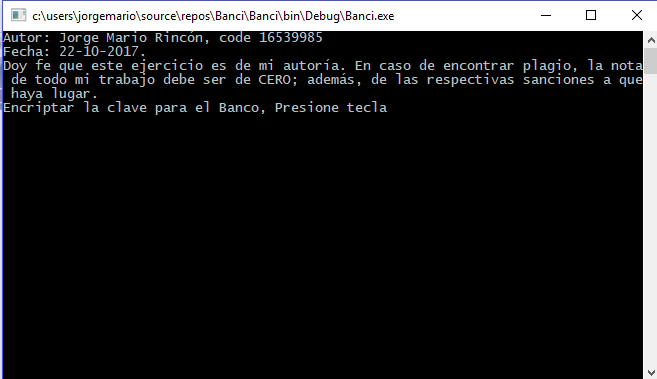
1982

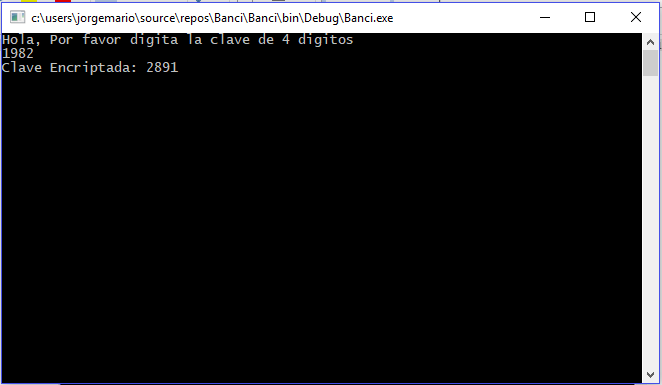
7898

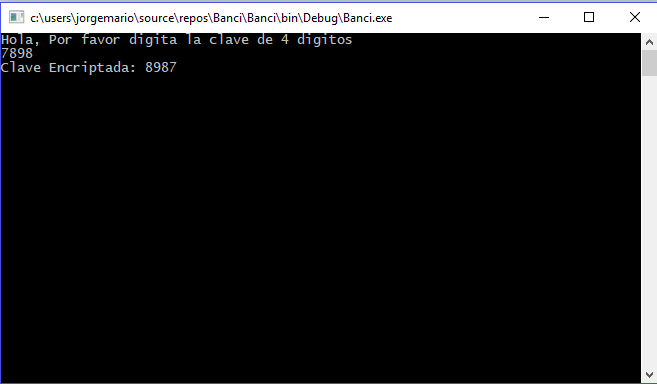
**Su programa Filtra valores inconsistentes Si / No**? No

No filtra entradas tipo STRING

**Resultado (captura imagen o imágenes del resultado (debe aparecer su nombre y fecha)**







|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Planteamiento del problema a resolver:  Un banco ha solicitado se diseñe un programa que permita encriptar la información de las contraseñas (4 números) digitada por teclado (como un único número) hasta el servidor principal, utilizando el siguiente criterio, el primer número se envía de último, el segundo, de penúltimo, el tercer número pasa a la segunda posición, el último pasa a ser primero: ejemplo: Sea 6532, se debe enviar como 2356 | | | | | | |
| Que datos no conozco y se deben solicitar (si se requiere más filas por favor anexarlas) | Que datos no conozco, pero los puedo calcular | Tipo  Variables requeridas | | Formulas a aplicar si se requieren | Requiere condicional (si / no) | Requiere ciclos (si / no) \_ cuales |
| Tipo | Variable |
| Fecha | Año bisiesto | int | a | Divisible entre 4, no divisible entre 100, divisible entre 400 | si | no |
|  |  | int | validar | Expresión regular para que se cumpla la fecha | si | no |
|  | Sustracción de la fecha | int | Dia, mes, a | Sumar días, meses, años | si | no |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

|  |
| --- |
| Restricciones a tener en cuenta – Filtros- |
| La fecha debe ser dia mes año separadas por comas o puntos o slash o guiones |
|  |
|  |

**Código**:

//Autor: Jorge Mario Rincón G., Código 16539985.

//Fecha: 23-10-2017.

//Doy fe que este ejercicio es de mi autoría. En caso de encontrar plagio, la nota de todo mi trabajo debe ser de CERO; además, de las respectivas sanciones a que haya lugar.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Text.RegularExpressions; // llamado de librerias para las expresiones regulares

namespace Miaplicacion

{

class SiguienteDiaProgram

{

static void Main(string[] args)

{

mensajeInicial();

string fecha = capturarFecha(); // definir fecha string y capturarla por teclado

// validar la fecha por puntos, o por slash o por guiones

Regex validar = new Regex(@"^(?:3[01]|[12][0-9]|0?[1-9])([\-/.])(0?[1-9]|1[1-2])\1\d{4}$");

// si es correcta la fecha prosige a evaluar el día siguiente

if (validar.IsMatch(fecha))

calcularDiaSiguiente(fecha); // método para calcular el sigueinte día

else

Console.Write("Fecha Mal Digitada"); // muestra mensaje de error

Console.ReadKey(); // espera presionar una tecla

}

public static void mensajeInicial()

{ // método publico para mostrar el mensaje de siempre

// mostrar el mensaje de siempre

Console.WriteLine("Autor: Jorge Mario Rincón, code 16539985");

Console.WriteLine("Fecha: 22-10-2017.");

Console.WriteLine("Doy fe que este ejercicio es de mi autoría. En caso de encontrar plagio, la nota de todo mi trabajo debe ser de CERO; además, de las respectivas sanciones a que haya lugar.");

Console.WriteLine("Mostrar el día siguiente:");

Console.WriteLine(""); //salto de linea

Console.ReadKey(); // sostener pantalla

Console.Clear(); // limpiar la consola

}

public static string capturarFecha() // método para capturar la fecha por el usuario

{

Console.WriteLine("Digita dia mes año, separados por comas o puntos o slash o guion");

Console.WriteLine("Ejemplo: ejemplo 02/03/2017");

Console.WriteLine("");

return Console.ReadLine(); // lee la fecha y la retorna

}

public static void calcularDiaSiguiente(string f) //método para calcular el siguiente día....

{

string caddia = f.Substring(0, 2); // sustraer el día

string cadmes = f.Substring(3, 2); //sustraer el mes

string cada = f.Substring(6, 4); //sustraer el año

int dia = Int32.Parse(caddia); // pasar el dia string a número

int mes = Int32.Parse(cadmes); // pasar el mes string a número

int a = Int32.Parse(cada); //pasar el año string a número

int bisiesto = 0; // variable para calcular el años biciesto

if ((a % 4 == 0) && !(a % 100 == 0) && (a % 400 == 0)) // si es biciesto es divisble por 4 , no por 100 si por 400

{

bisiesto = 1; // se activa biciesto

}

siguienteFecha(dia, mes, a, bisiesto); // calcula el siguiente dia, se le envian los parametros

}

public static void siguienteFecha(int dia, int mes, int a, int bisiesto)

{

//caso especial Febrero biciesto

if (mes == 2)

{

if (mes == 2 && dia == 28 && bisiesto == 1) // si es biciesto llega hasta 29

{

dia++;

}

else if (mes == 2 && dia == 29) // si no es biciesto pero dia 29, pasa al otro mes

{

mes++;

dia = 1;

}

else if (mes == 2 && dia == 28 && bisiesto == 0) // si no es biciesto pasa al otro mes

{

mes++;

dia = 1;

}

else if (mes == 2 && dia <= 27) // si es menor a 27 suma un dia en febrero

{

dia++;

}

}

else

{

// abril junio septiembre, noviembre son meses de 30 dias

if ((mes == 4) || (mes == 6) || (mes == 9) || (mes == 11))

{

if (dia == 30) // si es el final de mes pasa al otro

{

dia = 1;

mes++;

}

else if (dia <= 29) //menor a 29 dias suma un dia en el mismo mes

{

dia++;

}

}

else if (mes == 12 && dia == 31) // caso especial diciembre, dia 31 pasa de año

{

mes = 1;

dia = 1;

a++;

}

else // mes de enero, marzomayo, julio, agosto octubre, diciembre son de 31 dias

{

if ((mes == 1) || (mes == 3) || (mes == 5) || (mes == 7) || (mes == 8) || (mes == 10) || (mes == 12))

{

if (dia == 31) // pasa al siguiente mes

{

dia = 1;

mes++;

}

else if (dia <= 30) // aumenta un dia en el mismo mes

{

dia++;

}

}

}

}

// muestra la fecha

Console.Write("Siguiente Fecha: Dia " + dia + " Mes " + mes + " Año " + a);

}

}

}

**Usted probo el programa con diferentes valores Si / No**? Si

(Cuales):

28/02/2017

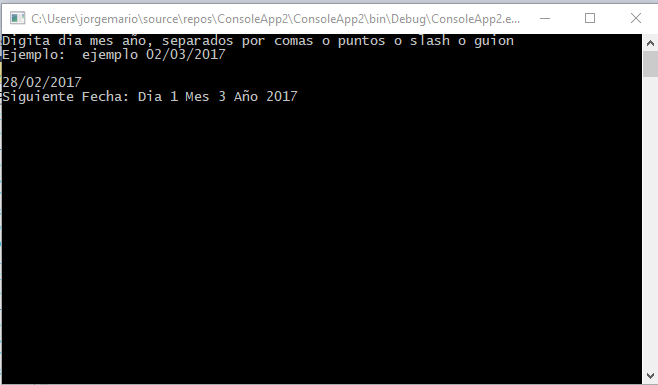
27/02/2017

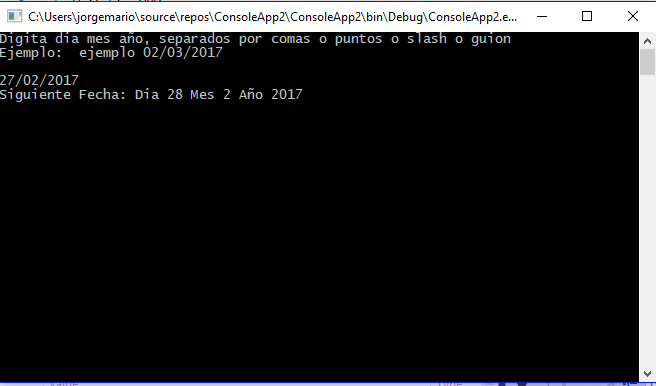
31/12/2017

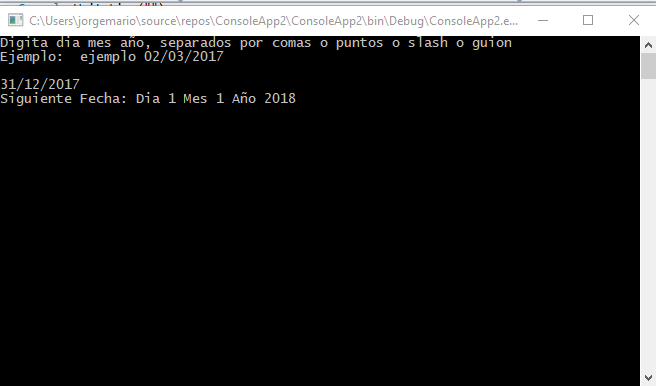
**Su programa Filtra valores inconsistentes Si / No**? No

No filtra fechas validas con bisiestos y día incoherentes ejemplo: 02/29/2017

**Resultado (captura imagen o imágenes del resultado (debe aparecer su nombre y fecha)**







3. Se lee un número de máximo tres dígitos (verifique que efectivamente sea de máximo tres dígitos) y se debe determinar si es un número capicúa, es decir, que leído de izquierda a derecha es igual que leído de derecha a izquierda. Por ejemplo: 727, 343, etc.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Planteamiento del problema a resolver:  3. Se lee un número de máximo tres dígitos (verifique que efectivamente sea de máximo tres dígitos) y se debe determinar si es un número capicúa, es decir, que leído de izquierda a derecha es igualque leído de derecha a izquierda. Por ejemplo: 727, 343, etc. | | | | | | |
| Que datos no conozco y se deben solicitar (si se requiere más filas por favor anexarlas) | Que datos no conozco, pero los puedo calcular | Tipo  Variables requeridas | | Formulas a aplicar si se requieren | Requiere condicional (si / no) | Requiere ciclos (si / no) \_ cuales |
| Tipo | Variable |
| Numero de tres cifras | Numero Leído al revés | int | num |  | Si de tres dígitos | no |
|  |  | String | numeroalreves |  | no | no |
|  |  | int | nuevonum | asignar |  |  |
|  |  | int | residuo | Capturar modulo |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

|  |
| --- |
| Filtro |
| Un número de tres dígitos |
|  |
|  |

**Código**:

//Autor: Jorge Mario Rincón G., Código 16539985.

//Fecha: 25-10-2017.

//Doy fe que este ejercicio es de mi autoría. En caso de encontrar plagio, la nota de todo mi trabajo debe ser de CERO; además, de las respectivas sanciones a que haya lugar.

//3. Se lee un número de máximo tres dígitos (verifique que efectivamente sea de máximo tres dígitos) y se debe determinar si es un número capicúa, es decir, que leído de izquierda a derecha es igual que leído de derecha a izquierda.Por ejemplo: 727, 343, etc.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp2

{

class Program

{

static void Main(string[] args)

{

mensajeInicial(); // mostrar mensaje de siempre

int num = capturarNumero(); // capturar el numero de tres cifras

if (num >= 100 && num <= 999) // validar si es de tres cifras

verificarCapicua(num); // si es de tres cifras validar si es capicúa

else

Console.Write("El numero debe ser de tres cifras, chao"); // el numero ingresado no es de tres cifras

Console.ReadKey();

}

public static void verificarCapicua(int n)

{

int nuevonum = n; // variable para guardar el nuvo numero mientras se divide entre 10 para obtener sus residuos acumulados y obtener el numero al réves

string numeroalreves= ""; // acumular el numero al revés

for (int residuo; nuevonum >= 10;)

{ // ciclo para dividir entre 10 hasta que no se pueda mas

residuo = nuevonum % 10; // se capturan los residuos

nuevonum = nuevonum / 10; // se capturan los cocientes

numeroalreves = numeroalreves + residuo; // se acumula el numero al revé

}

numeroalreves = numeroalreves+nuevonum; // unir todo el número

if (n == Int32.Parse(numeroalreves)) // comparar el número normal versus el número al revés

Console.Write("El número " + n + " Es Capicúa"); // mensaje el número es capicúa

else

Console.Write("El número no es CApicúa "+n+ " ,"+numeroalreves); // mensaje le número no es capicúa

}

// método para capturar el número

public static int capturarNumero()

{

Console.WriteLine("Digita un numero de tres digitos"); // digitar el número mensaje

return int.Parse(Console.ReadLine()); // capturar y retornar el número

}

public static void mensajeInicial()

{ // método publico para mostrar el mensaje de siempre

// mostrar el mensaje de siempre

Console.WriteLine("Autor: Jorge Mario Rincón, code 16539985");

Console.WriteLine("Fecha: 22-10-2017.");

Console.WriteLine("Doy fe que este ejercicio es de mi autoría. En caso de encontrar plagio, la nota de todo mi trabajo debe ser de CERO; además, de las respectivas sanciones a que haya lugar.");

Console.WriteLine("Numéro Capicúa, presiona tecla para continuar");

// Console.WriteLine(""); //salto de linea

Console.ReadKey(); // sostener pantalla

Console.Clear(); // limpiar la consola

}

}

}

**Usted probo el programa con diferentes valores Si / No**? Si

(Cuales):

456

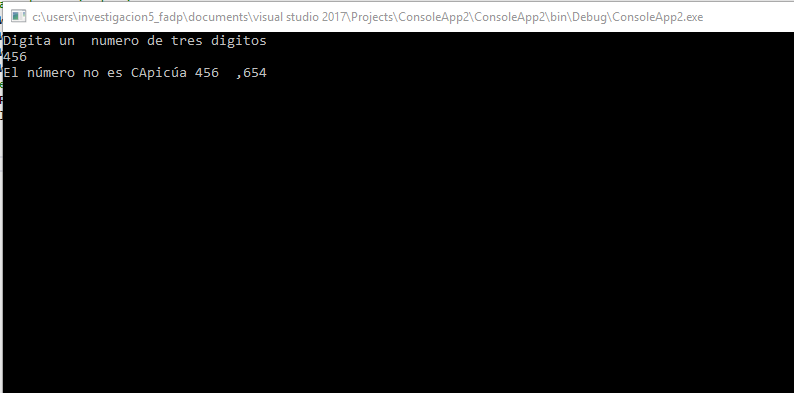
757

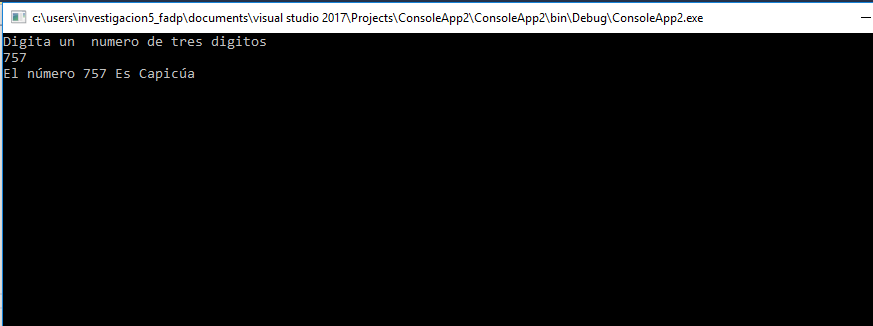
1

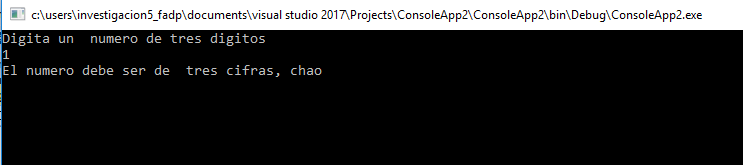
**Su programa Filtra valores inconsistentes Si / No**? No

No filtra las entradas String

**Resultado (captura imagen o imágenes del resultado (debe aparecer su nombre y fecha)**







|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Planteamiento del problema a resolver:  4. En una organización se tiene a los empleados agrupados por categoría, los de categoría 1 ganan $20.000, los de categoría 2, $15.000, los de categoría 3, $10.000 y los de categoría 4, $7.500. Se quiere un algoritmo que permita determinar cuánto debe pagarse a un empleado si se conoce el número de horas que trabajó durante el mes y la categoría a la que pertenece. Se sabe que a todos se les descuenta un 7.2% por concepto de salud, y si el salario total devengado (mensual) es menos de 1 ´000.000, se le da un subsidio del 15% sobre su salario mensual (sin descuentos). | | | | | | |
| Que datos no conozco y se deben solicitar (si se requiere más filas por favor anexarlas) | Que datos no conozco, pero los puedo calcular | Tipo  Variables requeridas | | Formulas a aplicar si se requieren | Requiere condicional (si / no) | Requiere ciclos (si / no) \_ cuales |
| Tipo | Variable |
| Número de horas laboradas | salud | Int[] horas trabajaas | Datos[] |  | si | no |
| Categoría a la que se aplica | subsidio | Int[] categoria | Datos[] |  | si | no |
|  | salariobruto | salariobruto | double | Horas\*valorcategoria | no | no |
|  |  |  |  |  |  |  |
|  | salariobrutosub | double | salariobrutosub |  | no | no |
|  | salariototal | dounle | salariototal |  | no | no |
|  | subsidio | double | subsidio | Salariobrutosub\*15% | si | no |
|  | Salud | Doublé | salud | Salariobruto\*7.2% | no | no |
|  | salario | double | salario | Salariobruto+subsidio-salud | no | no |
|  |  |  |  |  |  |  |

|  |
| --- |
| Filtro |
| Si se digita una categoría de 1 a 4 |
| Si se digita una categoría por fuera del rango |
|  |

**Código**:

//Autor: Jorge Mario Rincón G., Código 16539985.

//Fecha: 25-10-2017.

//Doy fe que este ejercicio es de mi autoría. En caso de encontrar plagio, la nota de todo mi trabajo debe ser de CERO; además, de las respectivas sanciones a que haya lugar.

/\*En una organización se tiene a los empleados agrupados por

categoría, los de categoría 1 ganan $20.000, los de categoría 2,

$15.000, los de categoría 3, $10.000 y los de categoría 4, $7.500.

Se

quiere un algoritmo que permita determinar cuánto debe pagarse a un

empleado si se conoce el número de horas que trabajó durante el mes

y la categoría a la que pertenece.

Se sabe que a todos se les descuenta

un 7.2% por concepto de salud, y si el salario total devengado

(mensual) es menos de 1 ´000.000, se le da un subsidio del 15% sobre

su salario mensual (sin descuentos).

\*/

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace EmpresaEjericio4

{

class Program

{

static void Main(string[] args)

{

mensajeInicial();

int[] categoria = { 20000, 15000, 10000, 7500 }; // se define un arreglo con las categorias

int[] determinar = capturarDatos(); // arreglo para capturar numero de horas y la categoria que pertenece

if (determinar[0] > 0 && determinar[0] <= 48 && determinar[1] >= 1 && determinar[1] <= 4) // validar los datos del usuario

calcularSalario(categoria, determinar); // método para calcular el salario total con descuentos e incrementos

else

Console.Write("digitaste numero de horas incorrecto o categoria no existe");

Console.ReadKey();

}

public static void calcularSalario(int[] categoria, int[] determinar) {

double salariobruto = determinar[0] \* categoria[determinar[1]-1]; // calcular el numero de horas \* la categoria - 1 por empezar el indice en cero

double subsidio=0,salariobrutosub=subsidio; // definir variables para el subsidio, salario bruto + el subsidio

if (salariobruto < 1000000) // si el salario devengado es menor a 1000000 tiene un subsidio del 15%

{

subsidio = (salariobruto \* 15) / 100; // calcular el 15% del subsidio

salariobrutosub = salariobruto + subsidio; //suma del salario bruto + el subsidio

}

double salud = (salariobruto \* 7.2) / 100; // calcular el porcentaje de la salud 7.2% del salario

double salario =salariobruto-salud; // descontar la salud al salario

Console.WriteLine("Número de Horas trabajadas: "+ determinar[0]); //mensaje mostrar numero de horas

Console.WriteLine("Categoria: " + determinar[1]+ ", Valor Categoria: " + categoria[determinar[1] - 1]); // //mensaje mostrar categoria y valor categoria

Console.WriteLine("Salario Bruto: " + salariobruto); // mostrar el slariobruto horas\*categoria

Console.WriteLine("Subsidio: " + subsidio); // mostrar el valor del subsidio

Console.WriteLine("salariobruto + subsidio " + salariobrutosub

); // mostrar el salario mas el subsidio

Console.WriteLine("Salud " + salud); // mostrar el valor de la salud

Console.WriteLine("Salario Final " + salario); // mostrar el slario total

}

public static int[] capturarDatos() {

int[] datos= new int[2]; // definir el arreglo para insertar las horas y la categoria

Console.WriteLine("Hola, digita el numero de horas"); // mostrar mensaje al usuario

datos[0] = int.Parse(Console.ReadLine()); // insertar las horas en el arreglo

Console.WriteLine("Digite la categoria de 1 a 4: "); // mostrar mensaje al usuario

Console.WriteLine("1. $20.000 2. $15.000 3.$10.000 4. $7500 "); // mostrar mensaje al usuario

datos[1] = int.Parse(Console.ReadLine()); // insertar en el arreglo la categoria

return datos; // devolver los datos

}

public static void mensajeInicial()

{ // método publico para mostrar el mensaje de siempre

// mostrar el mensaje de siempre

Console.WriteLine("Autor: Jorge Mario Rincón, code 16539985");

Console.WriteLine("Fecha: 25-10-2017.");

Console.WriteLine("Doy fe que este ejercicio es de mi autoría. En caso de encontrar plagio, la nota de todo mi trabajo debe ser de CERO; además, de las respectivas sanciones a que haya lugar.");

Console.WriteLine("Calcular Salario");

// Console.WriteLine(""); //salto de linea

Console.ReadKey(); // sostener pantalla

Console.Clear(); // limpiar la consola

}

}

}

**Usted probo el programa con diferentes valores Si / No**? Si

(Cuales):

10 y 1

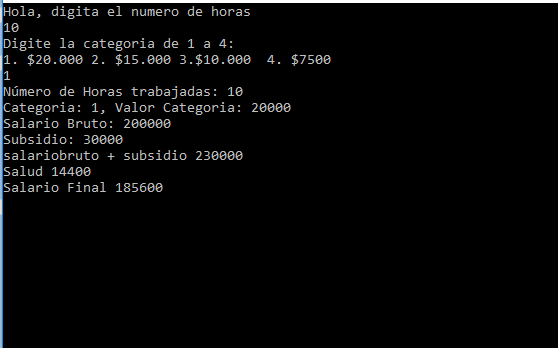
50 y 3

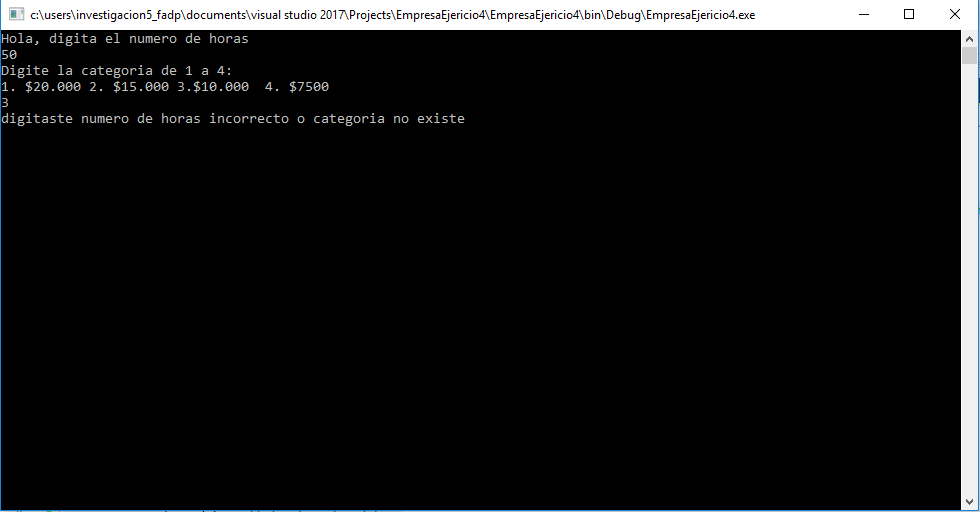
48 y 5

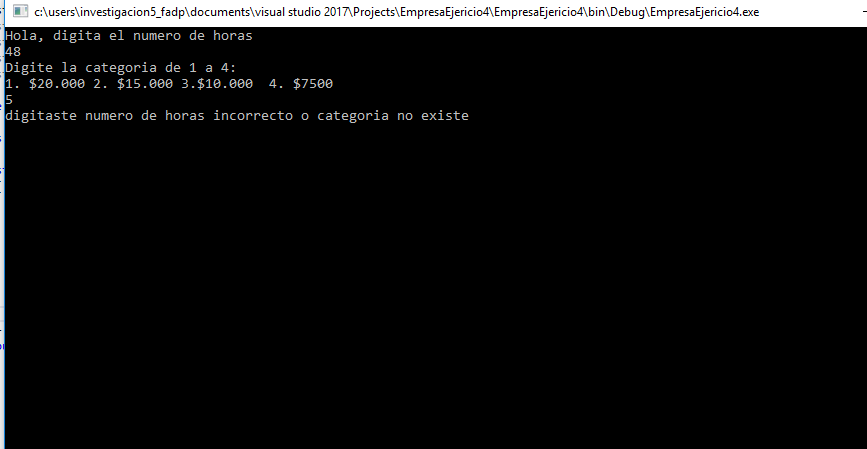
**Su programa Filtra valores inconsistentes Si / No**? No

No filtra un texto en el valor de horas

**Resultado (captura imagen o imágenes del resultado (debe aparecer su nombre y fecha)**







|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Planteamiento del problema a resolver:  5. Elabore un programa que tenga cuatro niveles de seguridad (documéntese sobre niveles de seguridad) para un programa, si el usuario logra ingresar imprimir el mensaje "Bienvenido", en caso contrario imprimir "Error clave" y el nivel del error | | | | | | |
| Que datos no conozco y se deben solicitar (si se requiere más filas por favor anexarlas) | Que datos no conozco, pero los puedo calcular | Tipo  Variables requeridas | | Formulas a aplicar si se requieren | Requiere condicional (si / no) | Requiere ciclos (si / no) \_ cuales |
| Tipo | Variable |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Planteamiento del problema a resolver:  6. Escriba un programa para calcular si un número es primo o no, recuerde que los números primos son aquellos que solo son divisibles por la unidad y por ellos mismos: ejemplo 5, 11, 17 | | | | | | |
| Que datos no conozco y se deben solicitar (si se requiere más filas por favor anexarlas) | Que datos no conozco, pero los puedo calcular | Tipo  Variables requeridas | | Formulas a aplicar si se requieren | Requiere condicional (si / no) | Requiere ciclos (si / no) \_ cuales |
| Tipo | Variable |
| Número para validar si es primo | Divisores exactos del número leido | int | primo | Primo%2==0 | si | Si saber sus antecesores divisibles |
|  |  | int | i | decremento |  |  |
|  |  | int | noprimo |  | si | no |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

|  |
| --- |
| Filtro |
| Validar si es un número positivo |
|  |
|  |

**Código**:

/\*

\* //Autor: Jorge Mario Rincón G., Código 16539985.

//Fecha: 25-10-2017.

//Doy fe que este ejercicio es de mi autoría. En caso de encontrar plagio, la nota de todo mi trabajo debe ser de CERO; además, de las respectivas sanciones a que haya lugar.

Escriba un programa para calcular si un número es primo o no,

recuerde que los números primos son aquellos que solo son divisibles

por la unidad y por ellos mismos: ejemplo 5, 11, 17

\*/

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp3

{

class Program

{

static void Main(string[] args)

{

mensajeInicial(); // mostrar mensaje inicial

int primo=capturarPrimo(); // capturar el numero primo

// validar si es numero

if (primo > 0)

{

validarPrimo(primo); // método para validar el número primo

}

else {

Console.Write("debe ser un número positivo..."); // mensaje si es numero

}

}

public static int capturarPrimo() { // método para capturar el primo

Console.WriteLine("Digita un número"); // mostrar mensaje

return int.Parse(Console.ReadLine()); // leer y devolver le número

}

public static void validarPrimo(int primo) { // método para validar el número

int noprimo = 0; // variable que valida el número primo

for (int i= primo-1; i>=2; i--) // ciclo para buscar más divisores y comprobar primo

{

if (primo % i == 0) // validar si es divisible

noprimo = 1; // i es divisible no es primo

}

if (noprimo == 0) // valida si es primo

Console.Write("El número " + primo + " Es Primo");

else {

Console.Write("El número " + primo + " NO Es Primo"); ;

}

Console.ReadKey();

}

public static void mensajeInicial()

{ // método publico para mostrar el mensaje de siempre

// mostrar el mensaje de siempre

Console.WriteLine("Autor: Jorge Mario Rincón, code 16539985");

Console.WriteLine("Fecha: 25-10-2017.");

Console.WriteLine("Doy fe que este ejercicio es de mi autoría. En caso de encontrar plagio, la nota de todo mi trabajo debe ser de CERO; además, de las respectivas sanciones a que haya lugar.");

Console.WriteLine("Calcular Primo");

// Console.WriteLine(""); //salto de linea

Console.ReadKey(); // sostener pantalla

Console.Clear(); // limpiar la consola

}

}

}

**Usted probo el programa con diferentes valores Si / No**? Si

(Cuales):

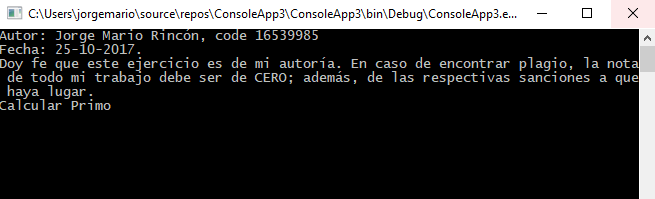
7

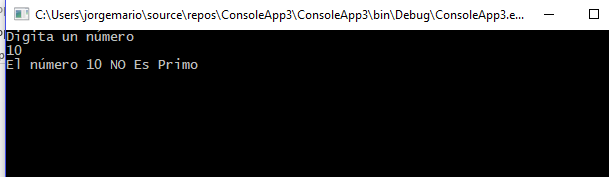
10

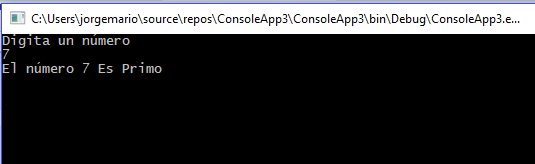
**Su programa Filtra valores inconsistentes Si / No**? No

No filtra un texto en el valor del número

**Resultado (captura imagen o imágenes del resultado (debe aparecer su nombre y fecha)**







|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Planteamiento del problema a resolver:  7. Lea los valores de los lados de un triángulo. Detecte y muestre un mensaje correspondiente a su tipo (EQUILÁTERO, ISÓSCELES, O ESCALENO) | | | | | | |
| Que datos no conozco y se deben solicitar (si se requiere más filas por favor anexarlas) | Que datos no conozco, pero los puedo calcular | Tipo  Variables requeridas | | Formulas a aplicar si se requieren | Requiere condicional (si / no) | Requiere ciclos (si / no) \_ cuales |
| Tipo | Variable |
| Lado1 |  | Int | Lado1 | Si son iguales | si | no |
| Lado2 |  | Int | Lado2 | Si son diferentes | si | no |
| Lado3 |  | Int | Lado3 | Dos lados iguales, diferente a otro | si | no |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

|  |
| --- |
| Restricciones a tener en cuenta – Filtros- |
|  |
|  |
|  |

**Código**:

//Fecha: 26-10-2017.

//Doy fe que este ejercicio es de mi autoría. En caso de encontrar plagio, la nota de todo mi trabajo debe ser de CERO; además, de las respectivas sanciones a que haya lugar.

/\*

\*

\*

\* Lea los valores de los lados de un triángulo. Detecte y muestre un

mensaje correspondiente a su tipo (EQUILÁTERO, ISÓSCELES, O

ESCALENO)

\* \*/

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp3

{

class Program

{

static void Main(string[] args)

{

mensajeInicial(); // mostrar mensaje inicial

int[] lados=leerLados(); // capturar lados del triangulo

validarTriangulo(lados); // método para validar el triangulo

}

public static void validarTriangulo(int[] lados)

{

if (lados[0] == lados[1] && lados[2] == lados[0]) // comparar si todos los lados son diferenes

{// validar si es equilatero

Console.WriteLine("El triangulo es equilátero: Lados iguales " + lados[0] + "," + lados[1] + "," + lados[2]); // mostra rmensaje

} else if (lados[0] != lados[1] && lados[0] != lados[2] && lados[1] != lados[2]) { // si todos los lados son diferentes

Console.WriteLine("El triangulo es Escaleno: Lados diferentes " + lados[0] + "," + lados[1] + "," + lados[2]); // mostrar mensaje

} else if ( ((lados[0] == lados[1]) && (lados[0]!=lados[2])) || ((lados[0]==lados[2]) && (lados[0] != lados[1])) ) { // si un lado es diferente a dos lados iguales

Console.WriteLine("El triangulo es Isósceles: dos lados iguales y uno diferente " + lados[0] + "," + lados[1] + "," + lados[2]); //mostrar mensaje

}

Console.ReadKey();

}

public static int[] leerLados() { // ,método para capturar los lados

int[] lados=new int[3]; // arreglo con los lados

Console.WriteLine("Hola por favor digita el primer lado del triangulo"); // Mostrar mensaje

lados[0] = int.Parse(Console.ReadLine()); //leer lado

Console.WriteLine("por favor digita el segundo lado del triangulo"); // Mostrar mensaje

lados[1] = int.Parse(Console.ReadLine()); //leer lado

Console.WriteLine("Por favor digita el Tercer lado del triangulo"); // Mostrar mensaje

lados[2] = int.Parse(Console.ReadLine()); //leer lado

return lados; // retornar el arreglo

}

public static void mensajeInicial()

{ // método publico para mostrar el mensaje de siempre

// mostrar el mensaje de siempre

Console.WriteLine("Autor: Jorge Mario Rincón, code 16539985");

Console.WriteLine("Fecha: 26-10-2017.");

Console.WriteLine("Doy fe que este ejercicio es de mi autoría. En caso de encontrar plagio, la nota de todo mi trabajo debe ser de CERO; además, de las respectivas sanciones a que haya lugar.");

Console.WriteLine("Tipos de Tirangulos");

Console.WriteLine(""); //salto de linea

Console.ReadKey(); // sostener pantalla

Console.Clear(); // limpiar la consola

}

}

}

**Usted probo el programa con diferentes valores Si / No**?

10 10 10

10 11 12

10 11 11

**Su programa Filtra valores inconsistentes Si / No**?

No filtra textos

**Resultado (captura imagen o imágenes del resultado (debe aparecer su nombre y fecha)**

