

Ejercicio Practica - Debugging Pseudo-codigo

Enunciado

Desarrollar un programa en el que se ingresen dos números enteros positivos y que genere y muestre un tercer numero que esté compuesto por las unidades del primer número y por las decenas del segundo.

Algoritmo Original

```
Programa Inicio

Entero Entrada_1, Entrada_2;

Imprimir << "Ingrese dos nros";
Leer >> Entrada_1 >> Entrada_2;

Entero Salida_nro = 0;

Salida_nro = Entrada_2[2] + Entrada_1[1];
Imprimir << "El nro resultante" << Salida_nro;

Fin
```

Algoritmo Reparado

```
Programa Inicio

Entero Entrada_1, Entrada_2;

Repetir

    Imprimir << "Ingrese dos nros";
    Leer >> Entrada_1 >> Entrada_2;

Mientras(!(Entrada_1 > 0 Y Entrada_2 > 9)) => (1)

Entero Salida_nro = 0;

Entero Salida_1 = Entrada_1 % 10;
Entero Salida_2 = ( Entrada_2 % 100 ) / 10; => (2)

Imprimir << "El nro resultante" << Salida_2 << Salida_1; => (3)

Salida_nro = Salida_2 * 10 + Salida_1;
Imprimir << "El nro resultante" << Salida_nro;
```

Fin

(1) Hasta que (Entrada_1 > 0 Y Entrada_2 > 9)

(2) Entero Salida_2 = (Entrada_2 / 10) % 10;

(3) Output alternativo

Guia 3 - Ejercicio 9

Enunciado

Leer un número y decidir si es primo. Imprimir "primo" si lo es o "siga participando" si no.

Inicio

Entero Numb

Escribir ("Ingrese un nro primo: ")

Leer (Numb)

Entero i = 2

Booleano Flag = Falso

Repetir

Si (Numb % i == 0) Entonces

Flag = !Flag (1)

Break

Fin Si

Mientras (i < N)

Si (Flag) Entonces

Imprimir ("Primo")

Sino

Imprimir ("Siga Participando")

Fin Si

Fin

(1) Flag = !Flag => Sobre escribe el valor del Flag negando el valor que tenia, ie:

Flag = false

Flag = !Flag

Flag = !false

Flag = true

Guia 3 - Ejercicio 13

Enunciado

Leer un string y contar la cantidad de vocales y de consonantes. Imprimir los resultados.

Resolución I

Inicio

Entero vowel_counter = 0

Entero consonant_counter = 0

Texto input_string

Escribir("Ingrese una frase u oración: ")

Leer(input_string)

Entero distance_lower_Capital = 'a' - 'A' (1)

Repetir

Caracter char_aux

Si (char_aux >= 'a' Y char_aux <= 'z') Entonces

char_aux = input_string[i] - distance_lower_Capital (2)

Si (char_aux >= 'A' Y char_aux <= 'Z') Entonces

Segun(char_aux)

Caso 'a'

Caso 'e'

Caso 'i'

Caso 'o'

Caso 'u'

vowel_counter++

De Otro Modo

consonant_counter++

Fin Segun

Fin Si

Mientras (i < Length(input_string) O input_string[i] == '\0')

Fin

(1) Calculamos la distancia entre 'A' y 'a'. 'A' = 65 y 'a' = 97 => 'a' - 'A' = 97 - 65 = 32 ☒

(2) Transformamos el caracter en minúscula a mayúscula.

[Link a tabla ASCII](#)

Resolución II

Inicio

```
Entero vowel_counter = 0
Entero consonant_counter = 0
```

```
Texto input_string
Escribir ( "Ingrese una frase u oración: " )
Leer ( input_string )
```

Repetir

Caracter char_aux

Si (char_aux >= 'A' Y char_aux <= 'Z') Entonces

Segun(char_aux)

Caso 'a'

Caso 'A'

Caso 'e'

Caso 'E'

Caso 'i'

Caso 'I'

Caso 'o'

Caso 'O'

Caso 'u'

Caso 'U'

vowel_counter++

De Otro Modo

consonant_counter++

Fin Segun

Fin Si

Mientras (i < Length(input_string) O input_string[i] == '\0')

Fin

Enunciado

Convertir una cantidad específica de días (ingresado por teclado) en años, semanas y días.

Ejemplo:

```
Input => Número de días: 1329
```

```
Output:
```

```
Años: 3
```

```
Semanas: 33
```

```
Días: 3
```

Resolución

```
Inicio
```

```
Entero days_input = 0
```

```
Escribir("Número de días: ")
```

```
Leer(days_input)
```

```
Escribir( "Years: ",    input_days / 365 )
```

```
Escribir( "Weeks: ",   ( input_days % 365 ) / 7 )
```

```
Escribir( "Days: " ,   ( input_days % 365 ) % 7 )
```

```
Fin
```

Guia 4 - Ejercicio 12

Enunciado

Leer 3 números flotantes y chequear si es posible hacer un triángulo con ellos. También calcular el perímetro del triángulo si los valores ingresados son válidos.

Resolución

Si ingresamos 3 aristas, debemos verificar que:

A + B > C

B + C > A

C + A > B

de cumplirse estas 3 condiciones, podemos formar un triangulo con ellos

```
Inicio
```

```
Entero Nro_I, Nro_II, Nro_III

Escribir ( "Ingrese 3 nros: " )
Leer ( Nro_I )
Leer ( Nro_II )
Leer ( Nro_III )

Si ( ( Nro_I + Nro_II > Nro_III ) Y ( Nro_II + Nro_III > Nro_I ) Y ( Nro_I +
Nro_III > Nro_II ) ) Entonces

    Imprimir ( "Se puede formar un Triangulo con: ", Nro_I, Nro_II, Nro_III )

Sino

    Imprimir ( "No se puede formar un Triangulo con: ", Nro_I, Nro_II, Nro_III
)

Fin Si

Fin
```

Guia 4 - Ejercicio 14

Enunciado

Leer un número entero entre 1 y 12 e imprimir el mes del año correspondiente (utilice enum).

Resolución

```
enum Months { Enero = 1, Febrero, Marzo, Abril, Mayo, Junio, Julio, Agosto
Septiembre, Octubre Noviembre, Diciembre };

Inicio

Entero month_input
Texto  month_output

Repetir

    Escribir ( "Ingrese nro de mes: " )
    Leer ( month_input )

Mientras ( !( month_input >= 1 && month_input <= 12 ) )

Segun ( month_input )

    Caso Enero
        month_output = "Enero"

    Caso Febrero
        month_output = "Febrero"
```

```
Caso Marzo
    month_output = "Marzo"

Caso Abril
    month_output = "Abril"

Caso Mayo
    month_output = "Mayo"

Caso Junio
    month_output = "Junio"

Caso Julio
    month_output = "Julio"

Caso Agosto
    month_output = "Agosto"

Caso Septiembre
    month_output = "Septiembre"

Caso Octubre
    month_output = "Octubre"

Caso Noviembre
    month_output = "Noviembre"

De Otro Modo
    month_output = "Diciembre"

Fin Segun

Imprimir ( month_output )

Fin
```