

PROGRAMACIÓN

Propuesta de resolución Trabajo Práctico N° 2 - Algoritmos

PUNTO 2

ALGORITMO Horas

ENTRADA: hora, minutos, segundos: enteros

SALIDA: suma_segundos: entero

V. AUXILIARES: error, minutos_a_segundos, horas_a_segundos: enteros

1. *Controlar datos*
2. *Calcular segundos totales*
3. ESCRIBIR (suma_segundos)
4. PARAR

1. Controlar datos

HACER

LEER(hora, minutos, segundos)

SI((hora>=0 ^ hora<=23) ^ (minutos>=0 ^ minutos <=59) ^ (segundos>=0 ^ segundos<=59))

ENTONCES

error <- 0 // datos correctos

SINO

error <- 1 //datos erróneos

ESCRIBIR ""ERROR: La hora es incorrecta"

FIN SI

MIENTRAS(error = 1)

2. Calcular segundos totales

minutos_a_segundos <- minutos * 60

horas_a_segundos <- hora * 60 * 60

suma_segundos <- minutos_a_segundos + horas_a_segundos + segundos

PROGRAMACIÓN

Propuesta de resolución Trabajo Práctico N° 2 - Algoritmos

PUNTO 3

ALGORITMO Escuela Rural

ENTRADA: transporte, distancia, hijos, desocupado, n: enteros

SALIDA: mensaje: cadena de caracteres

V. AUXILIARES: i: entero

1. *Analizar condicion alumnos*
2. PARAR

1. Analizar condición alumnos

LEER(n)

HACER n veces (i=1, ..., n)

 LEER(transporte)

 SEGUN(transporte)

 1: ESCRIBIR("Le corresponde \$1000")

 2: ESCRIBIR("Le corresponde \$500")

 3:

 LEER(distancia)

 SI(distancia>3)ENTONCES

 LEER(desocupado)

 SI(desocupado = 1)ENTONCES

 ESCRIBIR("Recibe plan Estudiar por \$5000")

 SINO

 LEER(hijos)

 SI(hijos>3)ENTONCES

 ESCRIBIR("Le corresponde \$3000")

 SINO

 ESCRIBIR("No le corresponde ayuda economica")

 SINO

 ESCRIBIR("No le corresponde ayuda economica")

FIN HACER

PUNTO 4: i.

ALGORITMO Cadena de Caracteres

ENTRADA: CC: char - (cadena de caracteres con MF=".")

SALIDA: total_vocales, cantidad_a, cantidad_e, cantidad_i, cantidad_o, cantidad_u: enteros

V. AUXILIARES:

1. *Inicializar*
2. *Procesar cadena*
3. ESCRIBIR (total_vocales, cantidad_a, cantidad_e, cantidad_i, cantidad_o, cantidad_u)
4. PARAR

1.Inicializar

cantidad_a <- 0

cantidad_e <- 0

cantidad_i <- 0

cantidad_o <- 0

cantidad_u <- 0

2. Procesar Cadena

LEER(CC)

MIENTRAS(CC <> MF)

 SEGÚN (CC)

 "a": cantidad_a <- cantidad_a + 1,

 "e": cantidad_e <- cantidad_e + 1,

 "i": cantidad_i <- cantidad_i + 1,

 "o": cantidad_o <- cantidad_o + 1,

 "u": cantidad_u <- cantidad_u + 1,

 FIN SEGÚN

 LEER (CC)

FIN MIENTRAS

total_vocales <- cantidad_a + cantidad_e + cantidad_i + cantidad_o + cantidad_u

PUNTO 4: ii. OPCIÓN 1**ALGORITMO Cadena de Caracteres**

ENTRADA: CC: char - (cadena de caracteres con MF=".")

SALIDA: CC: char, palabras: entero(+)

V. AUXILIARES: cantidad_a, cantidad_e, cantidad_i, cantidad_o, cantidad_u, vocales_distintas

1. Inicializar
2. Procesar cadena
3. ESCRIBIR (palabras)
4. PARAR

1.Inicializar

```
palabras <- 0
cantidad_a <- 0
cantidad_e <- 0
cantidad_i <- 0
cantidad_o <- 0
cantidad_u <- 0
vocales_distintas <- 0
```

2.Procesar cadena

LEER(CC)

MIENTRAS(CC <> MF)

 SI (CC <> " ") ENTONCES

 SEGÚN (CC)

 "a": cantidad_a <- cantidad_a + 1,

 "e": cantidad_e <- cantidad_e + 1,

 "i": cantidad_i <- cantidad_i + 1,

 "o": cantidad_o <- cantidad_o + 1,

 "u": cantidad_u <- cantidad_u + 1,

 FIN SEGÚN

 ESCRIBIR (CC)

 SINO

 ESCRIBIR (*)

//analizar los contadores

 SI (cantidad_a > 0) ENTONCES vocales_distintas <- vocales_distintas + 1

 SI (cantidad_e > 0) ENTONCES vocales_distintas <- vocales_distintas + 1

 SI (cantidad_i > 0) ENTONCES vocales_distintas <- vocales_distintas + 1

 SI (cantidad_o > 0) ENTONCES vocales_distintas <- vocales_distintas + 1

 SI (cantidad_u > 0) ENTONCES vocales_distintas <- vocales_distintas + 1

//controlar cuántas vocales distintas hubo en la palabra

 SI (vocales_distintas >= 3) ENTONCES palabras <- palabras +1

//inicializar contadores en 0

 LEER (CC)

FIN MIENTRAS

//analizar los contadores de la última palabra recorrida antes del punto. AGREGAR PASOS

//controlar vocales distintas. AGREGAR PASOS

//NOTA: en el código está completo

PUNTO 4: ii. OPCIÓN 2. (Con doble MIENTRAS)**ALGORITMO Cadena de Caracteres**ENTRADA: CC: char - (cadena de caracteres con MF="".")SALIDA: CC: char, palabras: entero(+)V. AUXILIARES: cantidad_a, cantidad_e, cantidad_i, cantidad_o, cantidad_u, vocales_distintas

1. Inicializar
2. Procesar cadena
3. ESCRIBIR (palabras)
4. PARAR

1.Inicializar

palabras <- 0

2.Procesar cadena

LEER(CC)

MIENTRAS(CC <> MF)

cantidad_a <- 0

cantidad_e <- 0

cantidad_i <- 0

cantidad_o <- 0

cantidad_u <- 0

vocales_distintas <- 0

MIENTRAS (CC <> “ “ ^ CC <> MF)

SEGÚN (CC)

‘a’: cantidad_a <- cantidad_a + 1,

‘e’: cantidad_e <- cantidad_e + 1,

‘i’: cantidad_i <- cantidad_i + 1,

‘o’: cantidad_o <- cantidad_o + 1,

‘u’: cantidad_u <- cantidad_u + 1,

FIN SEGÚN

ESCRIBIR (CC)

LEER (CC)

FIN MIENTRAS*//analizar los contadores (salió del mientras por ser espacio o punto)*

SI (cantidad_a > 0) ENTONCES vocales_distintas <- vocales_distintas + 1

SI (cantidad_e > 0) ENTONCES vocales_distintas <- vocales_distintas + 1

SI (cantidad_i > 0) ENTONCES vocales_distintas <- vocales_distintas + 1

SI (cantidad_o > 0) ENTONCES vocales_distintas <- vocales_distintas + 1

SI (cantidad_u > 0) ENTONCES vocales_distintas <- vocales_distintas + 1

//controlar cuántas vocales distintas hubo en la palabra

SI (vocales_distintas >= 3) ENTONCES palabras <- palabras + 1

//controlar si puedo escribir el último carácter leído y si puedo leer el próximo

SI (CC = “ ”) ENTONCES

ESCRIBIR (*)

LEER (CC)

FIN SI

FIN MIENTRAS

PUNTO 5

ALGORITMO Par de Números

ENTRADA: num1, num2: enteros

SALIDA: numeros, pares, suma_impares: entero(+)

V. AUXILIARES: i, aux: entero

1. Inicializar
2. Procesar números
3. ESCRIBIR (numeros, pares, suma_impares)
4. PARAR

1. Inicializar

```
suma_impares <- 0
```

```
pares <- 0
```

```
numeros <- 0
```

2. Procesar números

```
LEER(num1, num2)
```

```
//controlar si num1 es el mayor y ordenar
```

```
Si(num1 > num2)ENTONCES
```

```
    aux <- num1
```

```
    num1 <- num2
```

```
    num2 <- aux
```

```
FIN SI
```

```
HACER (i=num1, ..., num2)
```

```
    SI(i % 2 = 0) ENTONCES
```

```
        pares <- pares + 1
```

```
    SINO
```

```
        suma_impares <- suma_impares + i
```

```
FIN HACER
```

```
//calcular los números del rango
```

```
numeros <- (num2 - num1) + 1
```