

Tecnologías en Educación Matemática



MODULO 8

Dpto. de Ciencias e Ingeniería de la Computación
UNIVERSIDAD NACIONAL DEL SUR
Año 2019

1

Algoritmos

Problemas:

- ✓ Sumar los dígitos de un número N .
- ✓ Obtener el promedio de los dígitos de N .
- ✓ Sumar los dígitos de un número N que sean múltiplos de k .
- ✓ Sumar todos los números en el intervalo (a, b) que sean impares.
- ✓ Obtener el promedio de los números contenidos en el intervalo $(a, b]$.
- ✓ Obtener el promedio de los números contenidos en el intervalo $(a, b]$ que sean impares.
- ✓ Determinar si en el intervalo $(a, b]$ hay algún múltiplo de k .

2

Algoritmos

Calcular la suma de una secuencia de números ingresados por teclado, terminando en un num. negativo.

Secuencias de elementos



No sabemos cuántos elementos son

Algoritmos

Calcular la suma de una secuencia de números ingresados por teclado, terminando en un num. negativo.

ALGORITMO SumaSecuencia
DE: **Números ingresados por teclado**
DS: suma
Aux: N
COMIENZO
suma \leftarrow 0
Leer(N)
MIENTRAS (N>0)
 suma \leftarrow suma+N
 Leer(N)
FIN

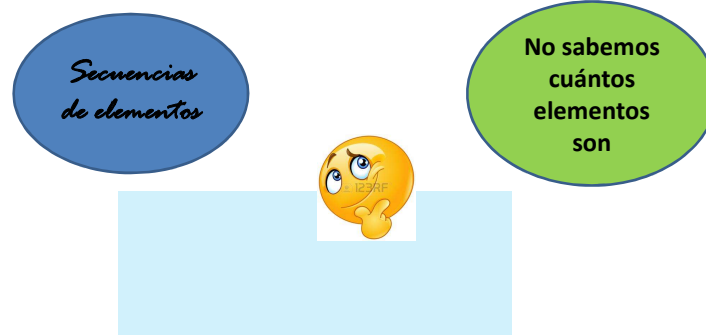
Secuencias de elementos



No sabemos cuántos elementos son

Algoritmos

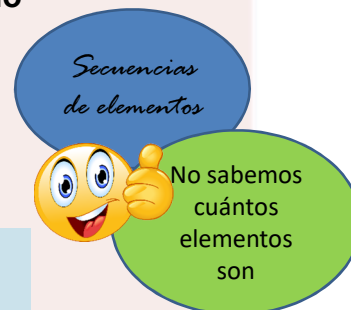
Obtener el máximo de una secuencia de números ingresados por teclado, terminando en un num. negativo.



Algoritmos

Obtener el máximo de una secuencia de números ingresados por teclado, terminando en un num. negativo.

```
ALGORITMO Maximo
DE: Números ingresados por teclado
DS: Max
Aux: N
COMIENZO
Max ← 0
Leer(N)
MIENTRAS (N>0)
    Si N>Max
        entonces Max ← N
Leer(N)
FIN
```



Algoritmos

Ver si un elemento aparece en una secuencia de números ingresados por teclado, terminando en un num. negativo.

Secuencias de elementos

No sabemos cuántos elementos son

Podemos no necesitar recorrerla completa



Algoritmos

Ver si un elemento aparece en una secuencia de números ingresados por teclado, terminando en un num. negativo.

ALGORITMO Esta
DE: elem, **Números ingresados por teclado**
DS: Esta
Aux: N
COMIENZO
Esta ← falso
Leer(N)
MIENTRAS (N>0) y (Esta=falso)
 Si N=elem
 entonces Esta← Verdadero
 sino **Leer(N)**
FIN

Secuencias de elementos

No sabemos cuántos elementos son

Podemos no necesitar recorrerla completa



Algoritmos . En resumen....

Recorrido genérico de una secuencia de elementos terminada en 0.

ALGORITMO ...

DE: **Números ingresados por teclado**

DS: Promedio

Aux: N, ...

COMIENZO

.....

Leer(N)

MIENTRAS (N<>0) ...

.....
Leer(N)

FIN

*Secuencias
de elementos*



No sabemos
cuántos
elementos
son

Algoritmos

Determinar si un elemento aparece en una secuencia, dada su longitud.

*Secuencias
de elementos*

Sabemos
cuántos
elementos
son.

Podemos
no
necesitar
recorrerla
completa



Algoritmos

Determinar si un elemento aparece en una secuencia, dada su longitud.

ALGORITMO Aparece

DE: Long, elem, **Números ing. por teclado**

DS: esta

Aux: N, i

COMIENZO

esta ← falso

i ← 1

Mientras (i ≤ Long) y (esta = falso)

Leer(N)

Si N = elem

entonces esta ← verdadero

i ← i + 1

*Secuencias
de elementos*

Sabemos
cuántos
elementos
son.

Podemos
no
necesitar
recorrerla
completa

Algoritmos Ciclos Anidados

Existen diversos problemas en los cuales es necesario utilizar ciclos repetitivos anidados, es decir, uno “dentro” del otro.

Supongamos el siguiente fragmento de código:

PARA i desde 1 hasta 5 hacer

...

PARA j desde 1 hasta 3

hacer

<acción I >

...

¿Cuántas veces
se repite la
acción I?

Algoritmos Ciclos Anidados

Problema

Contar la cantidad de pares de números naturales (a, b) cuyo producto sea múltiplo de un cierto valor K dado.
 a y b son números enteros que deben variar entre un límite inferior (LI) y un límite superior (LS) ingresados por el usuario.

Ejemplo:

Si $K=3$, $LI=2$, $LS=5$ el programa debe contar cada vez que el producto de los siguientes pares sea múltiplo de 3:

$(a, b) \rightarrow (2,2), (2,3), (2,4), (2,5), (3,2), (3,3), (3,4), (3,5),$
 $(4,2), (4,3), (4,4), (4,5), (5,2), (5,3), (5,4), (5,5)$

13

Algoritmos Ciclos Anidados

ALGORITMO anidados

DE: K, LI, LS

DS: cantidad

Daux: a, b

COMIENZO

cantidad $\leftarrow 0$

PARA a desde LI hasta LS hacer

PARA b desde LI hasta LS hacer

SI $((a*b) \bmod K)=0$

ENTONCES cantidad \leftarrow cantidad + 1

FIN



Algoritmos Problemas y Sub-problemas

Dada una secuencia de números ingresada por teclado, calcular la suma de aquellos cuya cantidad de dígitos es PAR. Inicialmente conocemos la longitud de la secuencia.

Problema: Recorrer una secuencia de números dada su longitud.

Sub-problema: Calcular la cantidad de dígitos de un número.



Ejemplo:

Long = 6

Secuencia de números

12 4 1111 457 3 32

Resultado = 1155

12 + 1111 + 32 (los números con cantidad par de dígitos)

Algoritmos Problemas y Sub-problemas

Problema: Dada una secuencia de números ingresada por teclado, calcular la suma de aquellos cuya cantidad de dígitos es PAR. Inicialmente contamos con la longitud de la secuencia.

```

ALGORITMO SumaSecuencia
DE: long, Números ingresados por teclado
DS: suma
Aux: N
COMIENZO
suma ← 0
Para i desde 1 hasta long
  Leer(N)
  {calcular la cant de dígitos de N}
  Si esa cantidad es PAR
  entonces
    suma ← suma + N
FIN
    
```

Primitiva
que resuelve
el sub-problema



Algoritmos Problemas y Sub-problemas

Problema: Un número N1 está contenido en dígitos en otro número N2 si todos los dígitos que aparecen en N1 también aparecen en N2, sin importar la cantidad de repeticiones ni el orden.

Por ejemplo:

N1 = 316 está contenido en dígitos en N2 = 76813

N1 = 3661 está contenido en dígitos en N2 = 1367

N1 = 397 está contenido en dígitos en N2 = 793

N1 = 3663 está contenido en dígitos en N2 = 63

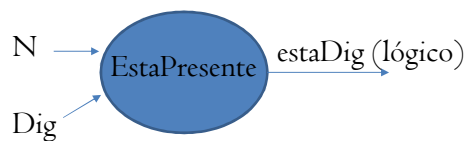
N1 = 397 NO está contenido en dígitos en N2 = 7988

Algoritmos Problemas y Sub-problemas

Subproblema: ¿Cómo determino si un dígito Dig está presente en el número N?

```

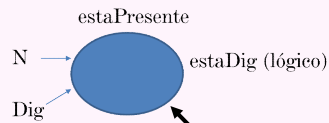
EstaDig ← falso
Mientras (EstaDig=falso) Y (N>0)  hacer
    SI (Dig = (N mod 10))
        ENTONCES
            EstaDig ← verdadero
        SINO
            N ← N div 10
    Fin Mientras
    
```



ALGORITMO Digos contenidos
 DE: Num1, Num2 {naturales}
 DS: Contenido {valor logico}
 Daux: Dig, NAux {naturales}, EstaDig {valor logico}
 COMIENZO
 Contenido ← verdadero
 MIENTRAS (Contenido=verdadero) Y (Num1>0) HACER
 Dig ← Num1 mod 10
 NAux ← Num2
 EstaDig ← falso
 MIENTRAS (EstaDig=falso) Y (NAux>0) HACER
 SI (Dig = (NAux mod 10))
 ENTONCES EstaDig ← verdadero
 SINO NAux ← NAux div 10
 Fin Mientras
 Si (EstaDig=falso)
 ENTONCES
 Contenido ← falso
 SINO
 Num1 ← Num1 div 10
 Fin Mientras
 FIN ALGORITMO



ALGORITMO Digos contenidos
 DE: Num1, Num2 {naturales}
 DS: Contenido {valor logico}
 Daux: Dig,
 COMIENZO
 Contenido ← verdadero
 MIENTRAS (Contenido=verdadero) Y (Num1>0) HACER
 Dig ← Num1 mod 10
 Si (estaPresente(Num2, Dig)=falso)
 ENTONCES
 Contenido ← falso
 SINO
 Num1 ← Num1 div 10
 Fin Mientras
 FIN ALGORITMO



Tecnologías en Educación Matemática



FIN MODULO 8

Dpto. de Ciencias e Ingeniería de la Computación
UNIVERSIDAD NACIONAL DEL SUR
Año 2019