

Índice

[**Introducción RursuS** 2](#_heading=h.gjdgxs)

[**Estructura de un programa en RursuS** 3](#_heading=h.30j0zll)

[**Conceptos Básicos RursuS** 4](#_heading=h.1fob9te)

[∙](#_heading=h.3znysh7) Variables en RursuS 4

[∙](#_heading=h.3dy6vkm) Tipos de Datos en RursuS 4

[∙](#_heading=h.2s8eyo1) Literales en RursuS 5

[**Operadores RursuS** 6](#_heading=h.17dp8vu)

[∙](#_heading=h.3rdcrjn) Operadores de Asignación y Aritméticos RursuS 6

[∙](#_heading=h.26in1rg) Operadores Unarios en RursuS 8

[∙](#_heading=h.lnxbz9) Operadores Igualdad y Relacionales en RursuS 9

[∙](#_heading=h.44sinio) Operadores Condicionales RursuS 12

[**Sentencias de Control** 13](#_heading=h.2xcytpi)

[∙](#_heading=h.1ci93xb) Sentencias de Decisión 13

[∙](#_heading=h.23ckvvd) Sentencias de Bucle 17

[∙](#_heading=h.3tbugp1) **Sentencias de ramificación** 20

[**Arrays RursuS** 22](#_heading=h.1mrcu09)

[**Tabla de precedencias** 25](#_heading=h.46r0co2)

[**Palabras reservadas** 27](#_heading=h.2lwamvv)

[**Algoritmos de conversión** 28](#_heading=h.111kx3o)

# **Introducción RursuS**

El lenguaje de programación RursuS fue creado por el profesor Kirstein Gatjens y estudiantes del Instituto Tecnológico de Costa Rica en el curso de Compiladores e Numerusérpretes durante el verano del 2022-23. La primera implementación del mismo la realizó Kirstein Gatjens sobre un computador con sistema operativo Linux. RursuS es el resultado de un proceso de desarrollo que comenzó con un lenguaje anterior, el Sator. El nombre del lenguaje RursuS es en latín que significa “Empezar de nuevo” y su extensión es “.RUR”.

El lenguaje RursuS es un lenguaje estructurado por bloques, en el mismo sentido que lo son otros lenguajes de programación tales como el lenguaje Pascal, el Ada o el Modula-2, es decir que es posible declarar subrutinas (pequeños trozos de programa) dentro de otras subrutinas. RursuS es un lenguaje tipado, pero muy flexible en cuanto al trabajo con los tipos; Posee un elemento separador el cual trabaja como un “Terminador” dicho separador es el punto(“.”)

El lenguaje RursuS posee un número reducido de 66 palabras reservadas que define el standard. Estas palabras reservadas pueden verse en la tabla siguiente:



# **Estructura de un programa en RursuS**

El lenguaje está estructurado en secciones. Estas secciones pueden ser opcionales, pero deben conservar su orden establecido.

Las secciones son:

* Sección de Título del programa (Nombre, Módulos)
* Sección de Constantes (tienen nombre y Tipo)
* Sección de Tipos
* Sección de Variables
* Sección de Prototipos de rutinas
* Sección de Rutinas
* Sección de programa principal

Para que un programa se ejecute debe tener una función “firmamentum” de lo contrario el compilador verifica si lo esta escrito se puede compilar, pero no se ejecutara el programa.

Ejemplo de las secciones del lenguaje

nomen INICIO. Sección de Título del programa

perpetuus Sección de constantes 

furibundus Sección de tipos 

commutabilis Sección de variables 

exemplar Sección de prototipos 

corpus Sección de rutinas 

firmamentum Sección de programa principal

initum

scribocatena (“Hola Mundo”) .

finis

.

# **Conceptos Básicos RursuS**

# [Variables en RursuS](https://www.manualweb.net/java/variables-java/)

Las variables RursuS son un espacio de memoria en el que guardamos un determinado valor (o dato). Para definir una variable seguiremos la estructura:

tipo\_variable identificador est valor .

RursuS es un lenguaje de tipado. Por lo cual todas las variables tendrán un tipo de dato (ya sea un tipo de dato primitivo), un nombre de identificador, est (que significa que es una variable ), opcional es valor de la variable y por último en terminador del lenguaje “.”. El tipo de dato se asignará a la hora de definir la variable. Además, todas las variables que se utilizan deben existir en la sección de variables.

Ejemplos de variables serían…

commutabilis

**numerus** number **est** 2 **.**

**catena** word **est** "Hola" **.**

**imago** letter **est** ‘M’ .

**dualis** flag **est** **true .**

## [Tipos de Datos en RursuS](https://www.manualweb.net/java/tipos-datos-primitivos-java/)

Como ya hemos comentado RursuS es un lenguaje de tipado. Es decir, se define el tipo de dato de la variable a la hora de definir esta. Es por ello que todas las variables tendrán un tipo de dato asignado.

El lenguaje RursuS da de base una serie de tipos de datos.

| Tipo de dato entero | numerus |
| --- | --- |
| Tipo de dato carácter | imago |
| Tipo de dato string | catena |
| Tipo de dato booleano | dualis |
| Tipo de dato conjunto FC | statuo |
| Tipo de dato archivo de texto FC | liber |
| Tipo de dato arreglos FC | ordo [**d1**] [**d2**] autem **tipo** |
| Tipo de dato registros FC | coniugo **d1. d2.** dixi |

## [Literales en RursuS](https://www.manualweb.net/java/literales-java/)

¿Qué son los literales RursuS?

Los valores literales son aquellos que podemos asignar a las variables. Dependiendo del tipo de variable podremos asignar unos valores u otros.

**Literales de enteros**

Los enteros que podemos utilizar serán  **numerus**. Los literales que les asignemos siempre será un número entero.

**numerus** variableNumerus est 12 **.**

**Literales de caracteres y cadenas**

Tanto los caracteres del tipo de dato **imago**, como las cadenas del tipo de datos **cadena** contienen caracteres.

Para los caracteres utilizaremos comillas simples para delimitarlos, mientras que para las cadenas utilizaremos comillas dobles.

**imago** variableImago est ‘a’ **.**

**cadena**  variableCadena est “cadena” **.**

Además, en las cadenas podemos utilizar una serie de secuencias de escape, las cuales empiezan por una barra invertida y siguen con un modificador:

| **Secuencia** | **Significado** |
| --- | --- |
| b | retroceso |
| t | tabular la cadena |
| n | salto de línea |
| f | form feed |
| r | retorno de carro |
| ' | comilla simple |
| " | comilla doble |
| \ | barra invertida |

# **Operadores RursuS**

## [Operadores de Asignación y Aritméticos RursuS](https://www.manualweb.net/java/operadores-asignacion-aritmeticos-java/)

Operador de Asignación

El operador en [RursuS](http://www.manualweb.net/tutorial-java/) más sencillo es el operador de asignación. Mediante este operador se asigna un valor a una variable. El operador de asignación es el símbolo de dos puntos seguido del símbolo igual.

La estructura del operador de asignación es:

variable := valor

Así podemos asignar valores a variables de tipo entero, cadena,…

numerus numero := 3.

catena cadena := "Hola Mundo".

dualis verdad := true.

Operadores Aritméticos

Los operadores aritméticos en [RursuS](http://www.manualweb.net/tutorial-java/) son los operadores que nos permiten realizar operaciones matemáticas: *suma, resta, multiplicación, división y resto*.

Los operadores aritméticos en [RursuS](http://www.manualweb.net/tutorial-java/) son:

| Operador | Descripción |
| --- | --- |
| + | Operador de Suma. Concatena cadenas para la suma de Catena |
| - | Operador de Resta |
| \* | Operador de Multiplicación |
| / | Operador de División |
| % | Operador de Residuo |

Los operadores aritméticos en [RursuS](http://www.manualweb.net/tutorial-java/) los utilizaremos entre dos literales o variables y el resultado, normalmente lo asignaremos a una variable o bien lo evaluamos.

variable = (valor1 aut variable1) operador (valor2 aut variable2).

Así podemos tener los siguientes usos en el caso de que queramos asignar su valor.

suma := 3 + 7 . *// Retorna 10*

resta := 5 - 2. *// Retorna 3*

multiplicación := 3 \* 2. *// Retorna 6*

división := 4 / 2. *// Retorna 2*

residuo := 5 % 3. *// Retorna 2*

Tomando en cuenta que pueden ser valores o variables:

suma = var + 3. *// Sumamos 3 al valor de la variable var*

resta = var - 4. *// Restamos 4 al valor de la variable var*

...

O podríamos utilizarlo en una condición

in (variable > suma + 3)

certus ...

En este caso no asignamos el resultado de la suma a una variable, solo lo evaluamos.

## [Operadores Unarios en RursuS](https://www.manualweb.net/java/operadores-unarios-java/)

**Operadores Unarios en RursuS**

Los operadores unarios en [RursuS](http://www.manualweb.net/tutorial-java/) son aquellos que solo requieren un operando para funcionar.

Los **operadores** unitarios que tenemos en [RursuS](http://www.manualweb.net/tutorial-java/) son:

| **Operador** | **Descripción** |
| --- | --- |
| + | Operador unario suma. Indica un número positivo. |
| - | Operador unario resta. Niega una expresión. |
| incrementum | Operador de incremento. Incrementa el valor en 1. |
| decrementum | Operador de decremento. Decrementa el valor en 1. |

Operadores unarios suma o resta

Los operadores unitarios de suma o resta son muy sencillos de utilizar. En el caso del operador unitario suma su uso es redundante. Con el operador unitario resta podemos invertir un valor.

Por ejemplo, podríamos tener el siguiente código:

**numerus** valor :**=** 2 **.**

**scribonumerus (-**valor**) .** *// Imprimirá por pantalla un -2*

**Operadores de incremento y decremento**

Los operadores de incremento se aplican solo como prefijo.

**incrementum** variable **.**

**decrementum** variable **.**

En todos los casos el valor de la variable acabará con una unidad más (para el operador de **incrementum**) o con una unidad menos (para el operador de **decrementum**).

Al utilizarlo como prefijo el valor de asignación será el valor del operando más el incremento de la unidad.

Es más sencillo verlo en código:

suma :**=** **incrementum** var **.**

Sería lo mismo que poner

var **:=** var **+** 1 **.**

suma :**=** var**.**

Exactamente lo mismo le sucede al operador de **decrementum**, pero restando una unidad.

## [Operadores Igualdad y Relacionales en RursuS](https://www.manualweb.net/java/operadores-igualdad-relacionales-java/)

Los operadores de igualdad y relacionales en [RursuS](http://www.manualweb.net/tutorial-java/) son aquellos que nos permiten comparar el contenido de una variable contra otra atendiendo a si son variables con un valor igual o diferentes o bien si los valores son mayores o menores.

El listado de operadores de igualdad y relacionales en [RursuS](http://www.manualweb.net/tutorial-java/) es:

| Operador | Descripción |
| --- | --- |
| = | igual a |
| **!=** | no igual a |
| > | mayor que |
| >= | mayor o igual que |
| < | menor que |
| <= | menor o igual que |

**Operadores de Igualdad**

Mediante los operadores de igualdad podemos comprobar si dos valores son iguales **(operador = )** o diferentes  **(operador**  **!=**  **)**.

La estructura de los **operadores de igualdad** es la siguiente:

var1 **=** var2

var1 **!=** var2

Podemos utilizar estos operadores de igualdad en [RursuS](http://www.manualweb.net/tutorial-java/) de la siguiente forma:

commutabilis

numerus var1 := 5.

numerus var2 := 3.

firmamentum

initum

in (var1 = var2)

certus scribocatena("Las variables son iguales").

in (var1 != var2)

certus scribocatena ("Las variables son diferentes ").

finis

.

**numerus** var1 :**=** 5**.**

**numerus** var2 :**=** 3**.**

**Operadores relacionales**

Permiten comprobar si un valor es mayor que **(operador >)**, menor que **(operador <)**, mayor o igual que **(>=)** y menor o igual que **(<=)**.

Al final el operador lo valida entre dos valores o variables con la estructura:

var1 **>** var2

var1 **<** var2

var1 **>=** var2

var1 **<=** var2

De esta forma podemos tener un código fuente que nos ayude a realizar estas validaciones de relación:

commutabilis

numerus var1 := 5.

numerus var2 := 3.

firmamentum

initum

in (var1 > var2)

certus scribocatena ("La variable 1 es mayor que la variable 2").

in (var1 < var2)

certus scribocatena ("La variable 1 es menor que la variable 2").

in (var1 >= var2)

certus scribocatena ("La variable 1 es mayor o igual que la variable 2").

in (var1 <= var2)

certus scribocatena ("La variable 1 es menor o igual que la variable 2").

finis

.

## [Operadores Condicionales RursuS](https://www.manualweb.net/java/operadores-condicionales-java/)

Los operadores condicionales en RursuS son aquellos que evalúan dos expresiones booleanas.

Dentro de los operadores condicionales en RursuS tenemos:

Operador Descripción

et Operador condicional AND

aut Operador condicional OR

xaunt Operador condicional XOR

non Operador condicional NOT

Operadores Condicionales

La estructura de los operadores condicionales en RursuS es:

(expresion\_booleana1 et expresion\_booleana2)

(expresion\_booleana1 aut expresion\_booleana2)

En el caso del operador condicional AND el resultado será veridicus y siempre y cuando las dos expresiones evaluadas sean veridicus. Si una de las expresiones es falsidicus el resultado de la expresión condicional AND será falsidicus.

Para el operador condicional OR el resultado será veridicus siempre que alguna de las dos expresiones sea veridicus.

En la compuerta NOT, el estado de la salida es inversa a la entrada. Evidentemente, una negación.

La XOR representa la función de la desigualdad, es decir, la salida es veridicus si las entradas no son iguales, de otro modo el resultado es falsidicus.

Podríamos ver el uso de los operadores condicionales en el siguiente código:

Commutabilis

numerus var1 := 5.

numerus var2 := 3.

firmamentum

initum

in ((var1 = 5) et (var2 =3))

certus scribocatena ("Las dos variables mantienen sus valores iniciales").

in ((var1 = 5) aut (var2 =3))

certus scribocatena("Al menos una variable mantiene su valor inicial").

finis

.

# **Sentencias de Control**

Un programa en [RursuS](http://www.manualweb.net/tutorial-java/) se ejecuta en orden desde la primera sentencia hasta la última. Si bien existen las sentencias de control de flujo las cuales permiten alterar el fujo de ejecución para tomar decisiones o repetir sentencias.

Dentro de las sentencias de control de flujo tenemos las siguientes:

* Sentencias de decisión
* Sentencias de bucle
* Sentencias de ramificación

## Sentencias de Decisión

Son sentencias que nos permiten tomar una decisión para poder ejecutar un bloque de sentencias u otro. Las sentencias de decisión son: in-certus-mentiri y aeger (if-then-else y switch en otros lenguajaes).

Mediante in-certus-mentiri podremos evaluar una decisión y elegir por un bloque u otro.

in (expresion)

certus . *// Bloque then*

mentiri

*// Bloque else*

.

Se evalúa la expresión indicada en la sentencia “in”. En el caso de que la expresión sea true se ejecutará el bloque de sentencias “then”  y en el caso de que la expresión sea false se ejecutará el bloque de sentencias ”else” .

La parte del else no tiene por qué existir. En este caso tendríamos una sentencia if-then

**in** **(**expresion**)**

certus . *// Bloque then*

**.**

**Aeger**

Para los casos en los que se tienen muchas ramas o caminos de ejecución en una sentencia “in” tenemos la sentencia “aeger”. La sentencia ” aeger” evalúa una expresión y ejecutará el bloque de sentencias que coincida con el valor de la expresión.

La estructura de la sentencia switch es:

aeger (expresion)

casus valor1:

bloque1.

neco.

casus valor2:

bloque2.

neco.

casus valor3:

bloque3.

neco.

detrimentum:

bloque\_por\_defecto.

.

Es importante ver que se utiliza la sentencia **neco.** La sentencia “neco” hace que se salga de la sentencia aeger y por lo tanto no se evalúe el resto de las sentencias. Por lo tanto, su uso es obligatorio al final de cada uno de los bloques.

Un ejemplo claro en el que podemos utilizar la sentencia aeger es para evaluar el valor de un mes en numérico y convertirlo a cadena. Este código quedaría de la siguiente forma:

commutabilis

**numerus** iMes **est** 3**.**

**catena** sMes **.**

firmamentum

initum

aeger **(**iMes**)**

casus 1**:**

sMes :**=** "Enero"**.**

**neco.**

**casus**  2**:**

sMes :**=** "Febrero"**.**

**neco.**

**casus**  3**:**

sMes :**=** "Marzo"**.**

**neco.**

**casus**  4**:**

sMes :**=** "Abril"**.**

**neco.**

**casus**  5**:**

sMes :**=** "Mayo"**.**

**neco.**

**casus**  6**:**

sMes :**=** "Junio"**.**

**neco.**

**casus**  7**:**

sMes :**=** "Julio"**.**

**neco.**

**casus**  8**:**

sMes :**=** "Agosto"**.**

**neco.**

**casus**  9**:**

sMes :**=** "Septiembre"**.**

**neco.**

**casus**  10**:**

sMes :**=** "Octubre"**.**

**neco.**

**casus**  11**:**

sMes :**=** "Noviembre"**.**

**neco.**

**casus**  12**:**

sMes :**=** "Diciembre"**.**

**neco.**

**detrimentum:**

sMes :**=** "Mes incorrecto"**.**

**.**

**scribocatena(**sMes**).**

finis

**.**

## Sentencias de Bucle

Las sentencias de bucle nos van a permitir ejecutar un bloque de sentencias tantas veces como queramos, o tantas veces como se cumpla una condición. En el momento que se cumpla esta condición será cuando salgamos del bucle. Las sentencias de bucle en [RursuS](http://www.manualweb.net/tutorial-java/) son: tempus, itero-usque y panis .

**Tempus**

La estructura repetitiva “tempus” realiza una primera evaluación antes de ejecutar el bloque. Si la expresión es “true” pasa a ejecutar de forma repetida el bloque de sentencias. Cada vez que termina de ejecutar el bloque de sentencias vuelve a evaluar la expresión. Si la expresión sigue siendo “true” vuelve a ejecutar el bloque. En el caso de que la expresión sea “false” se saldrá del bucle.

Es por ello que dentro del bloque de sentencias *deberán de existir sentencias que modifiquen la evaluación de la expresión, ya que de no hacerse se podría entrar en un bucle infinito*.

La estructura de la sentencia tempus es la siguiente:

tempus (expresion) opus

bloque\_sentencias.

.

Los casos de uso de una sentencia repetitiva tempus son variados, pero principalmente se utiliza para recorrer estructuras de datos o tener contadores.

Por ejemplo, podemos realizar un contador de 1 a 10 de la siguiente forma:

commutabilis

**numerus** contador **est** 1**.**

firmamentum

initum

tempus **(**contador **<=** 10**)** opus

**scribonumerus(**contador**).**

contador**+=1 .**

.

finis

**.**

**Itero-usque**

En el caso de la estructura repetitiva “itero-usque” el funcionamiento es el mismo que el de “aeger”. Pero con una diferencia, primero se ejecuta el bloque de sentencias y luego se evalúa la expresión. Por lo tanto, siempre se ejecutará, al menos una vez, el bloque de sentencias.

La estructura de la sentencia “itero-usque” es:

itero

bloque\_sentencias.

usque (expresion).

Al igual que anteriormente, en el bloque de sentencias deberemos de modificar alguna de las condiciones de la expresión para poder salir del bucle. Un ejemplo claro del bucle “itero-usque” sería el ejemplo en el que le pedimos al usuario que introduzca números por teclado, los cuales mostraremos en forma de eco por pantalla, hasta que introduzca el cero. En ese caso saldremos del bucle. Utilizaremos la estructura “itero-usque” en vez de la “aeger” ya que al menos vamos a pedirle al usuario un número.

El código sería el siguiente:

commutabilis

**numerus** iNumero**.**

firmamentum

initum

itero

**scribocatena(**"Introduce carácter por consola"**).**

iNumero :**=** lego**numerus().**

**scribonumerus(**iNumero**).**

usque **(**iNumero **>** 0**).**

finis

**.**

**Pannis**

Otra de las sentencias repetitivas que tenemos, a parte de los bucles “aeger” e itero-usque, es la sentencia pannis**.** La sentencia pannistiene la característica de que tiene bien definido el inicio del bloque, la evaluación de la expresión, el incremento de valor y el bloque de sentencias.

pannis (sentencias\_inicio (I :=0) expresion\_parada(auctum 10) expresion\_incremento (gradus 1) expresión\_cierre\_del\_ciclo (opus))

bloque\_sentencias.

.

Las funcionalidades en las que utilizaremos la sentencia pannisserán las mismas que las sentencias “aeger” e “itero-usque”, que serán contadores, recorrer estructuras,…

Si queremos definir un contador de 1 a 10 mediante una sentencia pannisutilizaremos el siguiente código:

pannis (I :=0 auctum 10 gradus 1 opus)

**scribocatena(**"Valor del contador: " **+** x**).**

.

En pantalla obtendremos el siguiente resultado:

Valor del contador: 1

Valor del contador: 2

Valor del contador: 3

Valor del contador: 4

Valor del contador: 5

Valor del contador: 6

Valor del contador: 7

Valor del contador: 8

Valor del contador: 9

Valor del contador: 10

## **Sentencias de ramificación**

Las sentencias de ramificación son aquellas que nos permiten romper con la ejecución lineal de un programa. El programa se va ejecutando de forma lineal, sentencia a sentencia. Si queremos romper esta linealidad tenemos las sentencias de ramificación. Las sentencias de ramificación en [RursuS](http://www.manualweb.net/tutorial-java/) son: **neco** y **pergo** .

En la sentencia selectiva “aeger” se utilizaba la sentencia “neco” para salir de las evaluaciones y así solo ejecutar el bloque de la opción correspondiente. Si bien podemos utilizar la sentencia “neco” con las sentencias repetitivas “aeger”, “itero-usque” y “pannis”. Cuando utilicemos el “neco” dentro de uno de estos bucles lo que se conseguirá es salirse de la ejecución del bucle hasta el siguiente bloque de sentencias. Mismo efecto que si la expresión de evaluación hubiese dado falsidicus (false).

Así podremos encontrarnos códigos como el siguiente:

tempus (expresion) opus

sentencia**(**s**).**

**neco.**

sentencias**(**s**).**

**.**

Al ejecutar la sentencia neco  ya no ejecutaremos las sentencias que vayan después.

El uso del “neco” dentro de estructuras repetitivas suele aparecer cuando estamos realizando la búsqueda de un elemento por una estructura de datos y lo hemos encontrado. Por ejemplo, si tenemos un array y queremos buscar un número dentro del array podríamos tener el siguiente código:

commutabilis

**ordo [8]** autem **numerus** numeros est **{**12**,**3**,**4**,**5**,**6**,**7**,**9**,**10**}.**

**dualis** encontrado **=** **falsidicus.**

**numerus** posicion **est 0.**

firmamentum

initum

tempus **(**posicion **<** quantus(numeros**))** opus

**in** **(**numeros**[**posicion**]** **=** 5**)**

certus encontrado :**=** **veridicus.**

**neco.**

posicion**+=1.**

**.**

**in** **(**encontrado**)**

certus **scribocatena(**"El número está en la posición: " **+** posicion**).**

mentiri

**scribocatena (**"Número no encontrado"**).**

finis

**.**

**Pergo**

Otra sentencia que podemos utilizar en los bucles es la sentencia “pergo”. A ejecutar una La sentencia “pergo” dejaremos de ejecutar las sentencias que quedan para acabar el bloque dentro de un bucle para volver a evaluar una expresión.

La estructura de una sentencia pergo sería:

**tempus** **(**expresion**)** **opus**

sentencia**(**s**)** iniciales**.**

**pergo.**

Sentencias**(**s**)** finales**.**

Al ejecutarse la sentencia pergo nunca se ejecutarán las sentencias finales.

# **Arrays RursuS**

¿Qué es un array en RursuS?

Un array [RursuS](http://www.manualweb.net/tutorial-java/) es una estructura de datos que nos permite almacenar una fila de datos de un mismo tipo. El tamaño de los arrays se declara en un primer momento y no puede cambiar en tiempo de ejecución como puede producirse en otros lenguajes. La declaración de un array en RursuS y su inicialización se realiza de la siguiente manera:

ordo **[tamaño]** autem tipo\_dato nombre\_array est {datos}**.**

Por ejemplo, podríamos declarar un array de caracteres e inicializarlo de la siguiente manera:

**ordo [8]** autem **numerus** numeros est **{**12**,**3**,**4**,**5**,**6**,**7**,**9**,**10**}.**

Los arrays [RursuS](http://www.manualweb.net/tutorial-java/) se numeran desde el elemento uno, que sería el primer elemento, hasta el tamaño que sería el último elemento. Para acceder a un elemento especifico utilizaremos los corchetes de la siguiente forma. Entendemos por acceso, tanto el intentar leer el elemento, como asignarle un valor.

arrayCaracteres**[**numero\_elemento**].**

Por ejemplo, para acceder al tercer elemento lo haríamos de la siguiente forma:

*// Lectura de su valor.*

**imago** x :**=** arrayCaracteres**[**3**].**

*// Asignación de un valor. Como se puede comprobar se pone el número tres, que coincide con el tercer elemento.*

arrayCaracteres**[**2**]** **=** 'b'**.**

El objeto array, aunque podríamos decir que no existe como tal, posee una variable, la cual podremos utilizar para facilitar su manejo.

Tamaño del array: quantus

Este atributo nos devuelve el número de elementos que posee el array. Hay que tener en cuenta que es una variable de solo lectura, es por ello que no podremos realizar una asignación a dicha variable. Por ejemplo, esto nos serviría a la hora de mostrar el contenido de los elementos de un array:

**ordo [3]** autem **numerus** numeros est **{**12**,**3**,**4**}.**

pannis (I :=0 auctum quantus(numeros) gradus 1 opus)

**scribonumerus** **(**numeros**[**I**]).**

.

Matrices o Arrays de varios subíndices

Podremos declarar arrays [RursuS](http://www.manualweb.net/tutorial-java/) de varios subíndices, pudiendo tener arrays [RursuS](http://www.manualweb.net/tutorial-java/) de dos niveles, que serían similares a las matrices, arrays [RursuS](http://www.manualweb.net/tutorial-java/) de tres niveles, que serían como cubos y así sucesivamente, si bien a partir del tercer nivel se pierde la perspectiva geométrica. Para declarar e inicializar un array de varios subíndices lo haremos de la siguiente manera:

ordo **[tamaño1] [tamaño2]** autem tipo\_dato nombre\_array est {{}}**.**

De esta forma podemos declarar una matriz [RursuS](http://www.manualweb.net/tutorial-java/) de 2x2 de la siguiente forma:

ordo [2][2] autem **numerus** A est {{1,2},{5,6 }} .

Nuestra lectura de los elementos de una matriz quedaría de la siguiente forma:

commutabilis

ordo [2][2] autem **numerus** A est {{1,2},{5,6 }} .

firmamentum

initum

pannis (x:=1 auctum quantus(numeros) gradus 1 opus)

pannis (y :=0 auctum quantus(numeros[x]) gradus 1 opus)

**scribonumerus(**matriz**[**x**][**y**]).**

**.**

**.**

finis

.

# **Tabla de precedencias**





# **Palabras reservadas**



# **Algoritmos de conversión**

<#--------------------------------------------

--------------Segmento de numerus-------------

----------------------------------------------#>

<# Tabla de equivalencia acii para numerus a imago #>

efficio ascii\_numerus\_to\_imago(numerus N) : imago.

initum

aeger n

initum

casus 0 opus reditus ''. neco.

casus 1 opus reditus '☺'. neco.

casus 2 opus reditus '☻'. neco.

casus 3 opus reditus '♥'. neco.

casus 4 opus reditus '♦'. neco.

casus 5 opus reditus '♣'. neco.

casus 6 opus reditus '♠'. neco.

casus 7 opus reditus '•'. neco.

casus 8 opus reditus '◘'. neco.

casus 9 opus reditus '○'. neco.

casus 10 opus reditus '◙'. neco.

casus 11 opus reditus '♂'. neco.

casus 12 opus reditus '♀'. neco.

casus 13 opus reditus '♪'. neco.

casus 14 opus reditus '♫'. neco.

casus 15 opus reditus '☼'. neco.

casus 16 opus reditus '►'. neco.

casus 17 opus reditus '◄'. neco.

casus 18 opus reditus '↕'. neco.

casus 19 opus reditus '‼'. neco.

casus 20 opus reditus '¶'. neco.

casus 21 opus reditus '§'. neco.

casus 22 opus reditus '▬'. neco.

casus 23 opus reditus '↨'. neco.

casus 24 opus reditus '↑'. neco.

casus 25 opus reditus '↓'. neco.

casus 26 opus reditus '→'. neco.

casus 27 opus reditus '←'. neco.

casus 28 opus reditus '∟'. neco.

casus 29 opus reditus '↔'. neco.

casus 30 opus reditus '▲'. neco.

casus 31 opus reditus '▼'. neco.

casus 32 opus reditus ' '. neco.

casus 33 opus reditus '!'. neco.

casus 34 opus reditus '"'. neco.

casus 35 opus reditus '#'. neco.

casus 36 opus reditus '$'. neco.

casus 37 opus reditus '%'. neco.

casus 38 opus reditus '&'. neco.

casus 39 opus reditus '\''. neco.

casus 40 opus reditus '('. neco.

casus 41 opus reditus ')'. neco.

casus 42 opus reditus '\*'. neco.

casus 43 opus reditus '+'. neco.

casus 44 opus reditus ','. neco.

casus 45 opus reditus '-'. neco.

casus 46 opus reditus '.'. neco.

casus 47 opus reditus '/'. neco.

casus 48 opus reditus '0'. neco.

casus 49 opus reditus '1'. neco.

casus 50 opus reditus '2'. neco.

casus 51 opus reditus '3'. neco.

casus 52 opus reditus '4'. neco.

casus 53 opus reditus '5'. neco.

casus 54 opus reditus '6'. neco.

casus 55 opus reditus '7'. neco.

casus 56 opus reditus '8'. neco.

casus 57 opus reditus '9'. neco.

casus 58 opus reditus ':'. neco.

casus 59 opus reditus ';'. neco.

casus 60 opus reditus '<'. neco.

casus 61 opus reditus '='. neco.

casus 62 opus reditus '>'. neco.

casus 63 opus reditus '?'. neco.

casus 64 opus reditus '@'. neco.

casus 65 opus reditus 'A'. neco.

casus 66 opus reditus 'B'. neco.

casus 67 opus reditus 'C'. neco.

casus 68 opus reditus 'D'. neco.

casus 69 opus reditus 'E'. neco.

casus 70 opus reditus 'F'. neco.

casus 71 opus reditus 'G'. neco.

casus 72 opus reditus 'H'. neco.

casus 73 opus reditus 'I'. neco.

casus 74 opus reditus 'J'. neco.

casus 75 opus reditus 'K'. neco.

casus 76 opus reditus 'L'. neco.

casus 77 opus reditus 'M'. neco.

casus 78 opus reditus 'N'. neco.

casus 79 opus reditus 'O'. neco.

casus 80 opus reditus 'P'. neco.

casus 81 opus reditus 'Q'. neco.

casus 82 opus reditus 'R'. neco.

casus 83 opus reditus 'S'. neco.

casus 84 opus reditus 'T'. neco.

casus 85 opus reditus 'U'. neco.

casus 86 opus reditus 'V'. neco.

casus 87 opus reditus 'W'. neco.

casus 88 opus reditus 'X'. neco.

casus 89 opus reditus 'Y'. neco.

casus 90 opus reditus 'Z'. neco.

casus 91 opus reditus '['. neco.

casus 92 opus reditus '\'. neco.

casus 93 opus reditus ']'. neco.

casus 94 opus reditus '^'. neco.

casus 95 opus reditus '\_'. neco.

casus 96 opus reditus '`'. neco.

casus 97 opus reditus 'a'. neco.

casus 98 opus reditus 'b'. neco.

casus 99 opus reditus 'c'. neco.

casus 100 opus reditus 'd'. neco.

casus 101 opus reditus 'e'. neco.

casus 102 opus reditus 'f'. neco.

casus 103 opus reditus 'g'. neco.

casus 104 opus reditus 'h'. neco.

casus 105 opus reditus 'i'. neco.

casus 106 opus reditus 'j'. neco.

casus 107 opus reditus 'k'. neco.

casus 108 opus reditus 'l'. neco.

casus 109 opus reditus 'm'. neco.

casus 110 opus reditus 'n'. neco.

casus 111 opus reditus 'o'. neco.

casus 112 opus reditus 'p'. neco.

casus 113 opus reditus 'q'. neco.

casus 114 opus reditus 'r'. neco.

casus 115 opus reditus 's'. neco.

casus 116 opus reditus 't'. neco.

casus 117 opus reditus 'u'. neco.

casus 118 opus reditus 'v'. neco.

casus 119 opus reditus 'w'. neco.

casus 120 opus reditus 'x'. neco.

casus 121 opus reditus 'y'. neco.

casus 122 opus reditus 'z'. neco.

casus 123 opus reditus '{'. neco.

casus 124 opus reditus '|'. neco.

casus 125 opus reditus '}'. neco.

casus 126 opus reditus 'Ï'. neco.

casus 127 opus reditus '⌂'. neco.

casus 128 opus reditus 'Ç'. neco.

casus 129 opus reditus 'ü'. neco.

casus 130 opus reditus 'é'. neco.

casus 131 opus reditus 'â'. neco.

casus 132 opus reditus 'ä'. neco.

casus 133 opus reditus 'à'. neco.

casus 134 opus reditus 'å'. neco.

casus 135 opus reditus 'ç'. neco.

casus 136 opus reditus 'ê'. neco.

casus 137 opus reditus 'ë'. neco.

casus 138 opus reditus 'è'. neco.

casus 139 opus reditus 'ï'. neco.

casus 140 opus reditus 'î'. neco.

casus 141 opus reditus 'ì'. neco.

casus 142 opus reditus 'Ä'. neco.

casus 143 opus reditus 'Å'. neco.

casus 144 opus reditus 'É'. neco.

casus 145 opus reditus 'æ'. neco.

casus 146 opus reditus 'Æ'. neco.

casus 147 opus reditus 'ô'. neco.

casus 148 opus reditus 'ö'. neco.

casus 149 opus reditus 'ò'. neco.

casus 150 opus reditus 'û'. neco.

casus 151 opus reditus 'ù'. neco.

casus 152 opus reditus 'ÿ'. neco.

casus 153 opus reditus 'Ö'. neco.

casus 154 opus reditus 'Ü'. neco.

casus 155 opus reditus 'ø'. neco.

casus 156 opus reditus '£'. neco.

casus 157 opus reditus 'Ø'. neco.

casus 158 opus reditus '×'. neco.

casus 159 opus reditus 'ƒ'. neco.

casus 160 opus reditus 'á'. neco.

casus 161 opus reditus 'í'. neco.

casus 162 opus reditus 'ó'. neco.

casus 163 opus reditus 'ú'. neco.

casus 164 opus reditus 'ñ'. neco.

casus 165 opus reditus 'Ñ'. neco.

casus 166 opus reditus 'ª'. neco.

casus 167 opus reditus 'º'. neco.

casus 168 opus reditus '¿'. neco.

casus 169 opus reditus '®'. neco.

casus 170 opus reditus '¬'. neco.

casus 171 opus reditus '½'. neco.

casus 172 opus reditus '¼'. neco.

casus 173 opus reditus '¡'. neco.

casus 174 opus reditus '«'. neco.

casus 175 opus reditus '»'. neco.

casus 176 opus reditus '░'. neco.

casus 177 opus reditus '▒'. neco.

casus 178 opus reditus '▓'. neco.

casus 179 opus reditus '│'. neco.

casus 180 opus reditus '┤'. neco.

casus 181 opus reditus 'Á'. neco.

casus 182 opus reditus 'Â'. neco.

casus 183 opus reditus 'À'. neco.

casus 184 opus reditus '©'. neco.

casus 185 opus reditus '╣'. neco.

casus 186 opus reditus '║'. neco.

casus 187 opus reditus '╗'. neco.

casus 188 opus reditus '╝'. neco.

casus 189 opus reditus '¢'. neco.

casus 190 opus reditus '¥'. neco.

casus 191 opus reditus '┐'. neco.

casus 192 opus reditus '└'. neco.

casus 193 opus reditus '┴'. neco.

casus 194 opus reditus '┬'. neco.

casus 195 opus reditus '├'. neco.

casus 196 opus reditus '─'. neco.

casus 197 opus reditus '┼'. neco.

casus 198 opus reditus 'ã'. neco.

casus 199 opus reditus 'Ã'. neco.

casus 200 opus reditus '╚'. neco.

casus 201 opus reditus '╔'. neco.

casus 202 opus reditus '╩'. neco.

casus 203 opus reditus '╦'. neco.

casus 204 opus reditus '╠'. neco.

casus 205 opus reditus '═'. neco.

casus 206 opus reditus '╬'. neco.

casus 207 opus reditus '¤'. neco.

casus 208 opus reditus 'ð'. neco.

casus 209 opus reditus 'Ð'. neco.

casus 210 opus reditus 'Ê'. neco.

casus 211 opus reditus 'Ë'. neco.

casus 212 opus reditus 'È'. neco.

casus 213 opus reditus 'ı'. neco.

casus 214 opus reditus 'Í'. neco.

casus 215 opus reditus 'Î'. neco.

casus 216 opus reditus 'Ï'. neco.

casus 217 opus reditus '┘'. neco.

casus 218 opus reditus '┌'. neco.

casus 219 opus reditus '█'. neco.

casus 220 opus reditus '▄'. neco.

casus 221 opus reditus '¦'. neco.

casus 222 opus reditus 'Ì'. neco.

casus 223 opus reditus '▀'. neco.

casus 224 opus reditus 'Ó'. neco.

casus 225 opus reditus 'ß'. neco.

casus 226 opus reditus 'Ô'. neco.

casus 227 opus reditus 'Ò'. neco.

casus 228 opus reditus 'õ'. neco.

casus 229 opus reditus 'Õ'. neco.

casus 230 opus reditus 'µ'. neco.

casus 231 opus reditus 'þ'. neco.

casus 232 opus reditus 'Þ'. neco.

casus 233 opus reditus 'Ú'. neco.

casus 234 opus reditus 'Û'. neco.

casus 235 opus reditus 'Ù'. neco.

casus 236 opus reditus 'ý'. neco.

casus 237 opus reditus 'Ý'. neco.

casus 238 opus reditus '¯'. neco.

casus 239 opus reditus '´'. neco.

casus 240 opus reditus ''. neco.

casus 241 opus reditus '±'. neco.

casus 242 opus reditus '‗'. neco.

casus 243 opus reditus 'À'. neco.

casus 244 opus reditus '¶'. neco.

casus 245 opus reditus '§'. neco.

casus 246 opus reditus '÷'. neco.

casus 247 opus reditus '¸'. neco.

casus 248 opus reditus '°'. neco.

casus 249 opus reditus '¨'. neco.

casus 250 opus reditus '·'. neco.

casus 251 opus reditus '¹'. neco.

casus 252 opus reditus '³'. neco.

casus 253 opus reditus '²'. neco.

casus 254 opus reditus '■'. neco.

casus 255 opus reditus ' '. neco.

detrimentum reditus ''. neco.

finis

.

finis

.

<# Pasa de numerus a imago #>

efficio numerus\_to\_imago(numerus N) : imago

initum

n := n % 256.

reditus ascii\_numerus\_to\_imago(n).

finis

.

<# Pasa de numerus a catena #>

efficio numerus\_to\_catena(numerus N) : catena

commutabilis

catena verbum est "".

dualis negans est falsidicus.

initum

in n < 0 certus

negans := veridicus.

n := n \* (-1).

.

tempus n > 0 opus

initum

temporalis = n % 10.

n /= 10.

aeger temporalis

initum

casus 0 opus verbum := "0" $+ verbum. neco.

casus 1 opus verbum := "1" $+ verbum. neco.

casus 2 opus verbum := "2" $+ verbum. neco.

casus 3 opus verbum := "3" $+ verbum. neco.

casus 4 opus verbum := "4" $+ verbum. neco.

casus 5 opus verbum := "5" $+ verbum. neco.

casus 6 opus verbum := "6" $+ verbum. neco.

casus 7 opus verbum := "7" $+ verbum. neco.

casus 8 opus verbum := "8" $+ verbum. neco.

casus 9 opus verbum := "9" $+ verbum. neco.

finis

.

finis

.

in negans certus

verbum := "-" $+ verbum.

.

reditus verbum.

finis

.

<# Pasa de numerus a dualis #>

efficio numerus\_to\_dualis(numerus N) : dualis.

initum

in N = 0

certus

reditus falsidicus.

mentiri

reditus veridicus.

.

finis

.

<# Pasa de numerus a statuo #>

efficio numerus\_to\_statuo(numerus N) : statuo

commutabilis

statuo exitus est {}.

numerus temporalis est 0.

initum

in n < 0 certus

exitus += '-'.

n := n \* (-1).

.

tempus n > 0 opus

initum

temporalis = n % 10.

n /= 10.

aeger temporalis

initum

casus 0 opus exitus += '0'. neco.

casus 1 opus exitus += '1'. neco.

casus 2 opus exitus += '2'. neco.

casus 3 opus exitus += '3'. neco.

casus 4 opus exitus += '4'. neco.

casus 5 opus exitus += '5'. neco.

casus 6 opus exitus += '6'. neco.

casus 7 opus exitus += '7'. neco.

casus 8 opus exitus += '8'. neco.

casus 9 opus exitus += '9'. neco.

finis

.

exitus += numerus\_to\_imago(temporalis).

finis

.

finis

.

<# Pasa de numerus a liber #>

efficio numerus\_to\_liber(numerus N) : liber

commutabilis

catena n\_catena est numerus\_to\_catena(n).

liber fi est {@ "c:\" $+ numerus\_to\_catena(n) $+ ".txt" , D @}.

initum

scripturam(fi, n\_catena).

prope(fi).

finis

.

<#--------------------------------------------

--------------Segmento de imagos--------------

----------------------------------------------#>

<# Tabla de equivalencia acii para imago a numerus #>

efficio ascii\_imago\_to\_numerus(imago cht) : numerus

initum

aeger cht

initum

casus '' opus reditus 0. neco.

casus '☺' opus reditus 1. neco.

casus '☻' opus reditus 2. neco.

casus '♥' opus reditus 3. neco.

casus '♦' opus reditus 4. neco.

casus '♣' opus reditus 5. neco.

casus '♠' opus reditus 6. neco.

casus '•' opus reditus 7. neco.

casus '◘' opus reditus 8. neco.

casus '○' opus reditus 9. neco.

casus '◙' opus reditus 10. neco.

casus '♂' opus reditus 11. neco.

casus '♀' opus reditus 12. neco.

casus '♪' opus reditus 13. neco.

casus '♫' opus reditus 14. neco.

casus '☼' opus reditus 15. neco.

casus '►' opus reditus 16. neco.

casus '◄' opus reditus 17. neco.

casus '↕' opus reditus 18. neco.

casus '‼' opus reditus 19. neco.

casus '¶' opus reditus 20. neco.

casus '§' opus reditus 21. neco.

casus '▬' opus reditus 22. neco.

casus '↨' opus reditus 23. neco.

casus '↑' opus reditus 24. neco.

casus '↓' opus reditus 25. neco.

casus '→' opus reditus 26. neco.

casus '←' opus reditus 27. neco.

casus '∟' opus reditus 28. neco.

casus '↔' opus reditus 29. neco.

casus '▲' opus reditus 30. neco.

casus '▼' opus reditus 31. neco.

casus ' ' opus reditus 32. neco.

casus '!' opus reditus 33. neco.

casus '"' opus reditus 34. neco.

casus '#' opus reditus 35. neco.

casus '$' opus reditus 36. neco.

casus '%' opus reditus 37. neco.

casus '&' opus reditus 38. neco.

casus '\'' opus reditus 39. neco.

casus '(' opus reditus 40. neco.

casus ')' opus reditus 41. neco.

casus '\*' opus reditus 42. neco.

casus '+' opus reditus 43. neco.

casus ',' opus reditus 44. neco.

casus '-' opus reditus 45. neco.

casus '.' opus reditus 46. neco.

casus '/' opus reditus 47. neco.

casus '0' opus reditus 48. neco.

casus '1' opus reditus 49. neco.

casus '2' opus reditus 50. neco.

casus '3' opus reditus 51. neco.

casus '4' opus reditus 52. neco.

casus '5' opus reditus 53. neco.

casus '6' opus reditus 54. neco.

casus '7' opus reditus 55. neco.

casus '8' opus reditus 56. neco.

casus '9' opus reditus 57. neco.

casus ':' opus reditus 58. neco.

casus ';' opus reditus 59. neco.

casus '<' opus reditus 60. neco.

casus '=' opus reditus 61. neco.

casus '>' opus reditus 62. neco.

casus '?' opus reditus 63. neco.

casus '@' opus reditus 64. neco.

casus 'A' opus reditus 65. neco.

casus 'B' opus reditus 66. neco.

casus 'C' opus reditus 67. neco.

casus 'D' opus reditus 68. neco.

casus 'E' opus reditus 69. neco.

casus 'F' opus reditus 70. neco.

casus 'G' opus reditus 71. neco.

casus 'H' opus reditus 72. neco.

casus 'I' opus reditus 73. neco.

casus 'J' opus reditus 74. neco.

casus 'K' opus reditus 75. neco.

casus 'L' opus reditus 76. neco.

casus 'M' opus reditus 77. neco.

casus 'N' opus reditus 78. neco.

casus 'O' opus reditus 79. neco.

casus 'P' opus reditus 80. neco.

casus 'Q' opus reditus 81. neco.

casus 'R' opus reditus 82. neco.

casus 'S' opus reditus 83. neco.

casus 'T' opus reditus 84. neco.

casus 'U' opus reditus 85. neco.

casus 'V' opus reditus 86. neco.

casus 'W' opus reditus 87. neco.

casus 'X' opus reditus 88. neco.

casus 'Y' opus reditus 89. neco.

casus 'Z' opus reditus 90. neco.

casus '[' opus reditus 91. neco.

casus '\' opus reditus 92. neco.

casus ']' opus reditus 93. neco.

casus '^' opus reditus 94. neco.

casus '\_' opus reditus 95. neco.

casus '`' opus reditus 96. neco.

casus 'a' opus reditus 97. neco.

casus 'b' opus reditus 98. neco.

casus 'c' opus reditus 99. neco.

casus 'd' opus reditus 100. neco.

casus 'e' opus reditus 101. neco.

casus 'f' opus reditus 102. neco.

casus 'g' opus reditus 103. neco.

casus 'h' opus reditus 104. neco.

casus 'i' opus reditus 105. neco.

casus 'j' opus reditus 106. neco.

casus 'k' opus reditus 107. neco.

casus 'l' opus reditus 108. neco.

casus 'm' opus reditus 109. neco.

casus 'n' opus reditus 110. neco.

casus 'o' opus reditus 111. neco.

casus 'p' opus reditus 112. neco.

casus 'q' opus reditus 113. neco.

casus 'r' opus reditus 114. neco.

casus 's' opus reditus 115. neco.

casus 't' opus reditus 116. neco.

casus 'u' opus reditus 117. neco.

casus 'v' opus reditus 118. neco.

casus 'w' opus reditus 119. neco.

casus 'x' opus reditus 120. neco.

casus 'y' opus reditus 121. neco.

casus 'z' opus reditus 122. neco.

casus '{' opus reditus 123. neco.

casus '|' opus reditus 124. neco.

casus '}' opus reditus 125. neco.

casus 'Ï' opus reditus 126. neco.

casus '⌂' opus reditus 127. neco.

casus 'Ç' opus reditus 128. neco.

casus 'ü' opus reditus 129. neco.

casus 'é' opus reditus 130. neco.

casus 'â' opus reditus 131. neco.

casus 'ä' opus reditus 132. neco.

casus 'à' opus reditus 133. neco.

casus 'å' opus reditus 134. neco.

casus 'ç' opus reditus 135. neco.

casus 'ê' opus reditus 136. neco.

casus 'ë' opus reditus 137. neco.

casus 'è' opus reditus 138. neco.

casus 'ï' opus reditus 139. neco.

casus 'î' opus reditus 140. neco.

casus 'ì' opus reditus 141. neco.

casus 'Ä' opus reditus 142. neco.

casus 'Å' opus reditus 143. neco.

casus 'É' opus reditus 144. neco.

casus 'æ' opus reditus 145. neco.

casus 'Æ' opus reditus 146. neco.

casus 'ô' opus reditus 147. neco.

casus 'ö' opus reditus 148. neco.

casus 'ò' opus reditus 149. neco.

casus 'û' opus reditus 150. neco.

casus 'ù' opus reditus 151. neco.

casus 'ÿ' opus reditus 152. neco.

casus 'Ö' opus reditus 153. neco.

casus 'Ü' opus reditus 154. neco.

casus 'ø' opus reditus 155. neco.

casus '£' opus reditus 156. neco.

casus 'Ø' opus reditus 157. neco.

casus '×' opus reditus 158. neco.

casus 'ƒ' opus reditus 159. neco.

casus 'á' opus reditus 160. neco.

casus 'í' opus reditus 161. neco.

casus 'ó' opus reditus 162. neco.

casus 'ú' opus reditus 163. neco.

casus 'ñ' opus reditus 164. neco.

casus 'Ñ' opus reditus 165. neco.

casus 'ª' opus reditus 166. neco.

casus 'º' opus reditus 167. neco.

casus '¿' opus reditus 168. neco.

casus '®' opus reditus 169. neco.

casus '¬' opus reditus 170. neco.

casus '½' opus reditus 171. neco.

casus '¼' opus reditus 172. neco.

casus '¡' opus reditus 173. neco.

casus '«' opus reditus 174. neco.

casus '»' opus reditus 175. neco.

casus '░' opus reditus 176. neco.

casus '▒' opus reditus 177. neco.

casus '▓' opus reditus 178. neco.

casus '│' opus reditus 179. neco.

casus '┤' opus reditus 180. neco.

casus 'Á' opus reditus 181. neco.

casus 'Â' opus reditus 182. neco.

casus 'À' opus reditus 183. neco.

casus '©' opus reditus 184. neco.

casus '╣' opus reditus 185. neco.

casus '║' opus reditus 186. neco.

casus '╗' opus reditus 187. neco.

casus '╝' opus reditus 188. neco.

casus '¢' opus reditus 189. neco.

casus '¥' opus reditus 190. neco.

casus '┐' opus reditus 191. neco.

casus '└' opus reditus 192. neco.

casus '┴' opus reditus 193. neco.

casus '┬' opus reditus 194. neco.

casus '├' opus reditus 195. neco.

casus '─' opus reditus 196. neco.

casus '┼' opus reditus 197. neco.

casus 'ã' opus reditus 198. neco.

casus 'Ã' opus reditus 199. neco.

casus '╚' opus reditus 200. neco.

casus '╔' opus reditus 201. neco.

casus '╩' opus reditus 202. neco.

casus '╦' opus reditus 203. neco.

casus '╠' opus reditus 204. neco.

casus '═' opus reditus 205. neco.

casus '╬' opus reditus 206. neco.

casus '¤' opus reditus 207. neco.

casus 'ð' opus reditus 208. neco.

casus 'Ð' opus reditus 209. neco.

casus 'Ê' opus reditus 210. neco.

casus 'Ë' opus reditus 211. neco.

casus 'È' opus reditus 212. neco.

casus 'ı' opus reditus 213. neco.

casus 'Í' opus reditus 214. neco.

casus 'Î' opus reditus 215. neco.

casus 'Ï' opus reditus 216. neco.

casus '┘' opus reditus 217. neco.

casus '┌' opus reditus 218. neco.

casus '█' opus reditus 219. neco.

casus '▄' opus reditus 220. neco.

casus '¦' opus reditus 221. neco.

casus 'Ì' opus reditus 222. neco.

casus '▀' opus reditus 223. neco.

casus 'Ó' opus reditus 224. neco.

casus 'ß' opus reditus 225. neco.

casus 'Ô' opus reditus 226. neco.

casus 'Ò' opus reditus 227. neco.

casus 'õ' opus reditus 228. neco.

casus 'Õ' opus reditus 229. neco.

casus 'µ' opus reditus 230. neco.

casus 'þ' opus reditus 231. neco.

casus 'Þ' opus reditus 232. neco.

casus 'Ú' opus reditus 233. neco.

casus 'Û' opus reditus 234. neco.

casus 'Ù' opus reditus 235. neco.

casus 'ý' opus reditus 236. neco.

casus 'Ý' opus reditus 237. neco.

casus '¯' opus reditus 238. neco.

casus '´' opus reditus 239. neco.

casus '' opus reditus 240. neco.

casus '±' opus reditus 241. neco.

casus '‗' opus reditus 242. neco.

casus 'À' opus reditus 243. neco.

casus '¶' opus reditus 244. neco.

casus '§' opus reditus 245. neco.

casus '÷' opus reditus 246. neco.

casus '¸' opus reditus 247. neco.

casus '°' opus reditus 248. neco.

casus '¨' opus reditus 249. neco.

casus '·' opus reditus 250. neco.

casus '¹' opus reditus 251. neco.

casus '³' opus reditus 252. neco.

casus '²' opus reditus 253. neco.

casus '■' opus reditus 254. neco.

casus ' ' opus reditus 255. neco.

detrimentum reditus 0. neco.

finis

.

finis

.

<# Tabla de equivalencia acii para imago a catena #>

efficio ascii\_imago\_to\_catena(imago cht) : catena

initum

aeger cht

initum

casus ''. opus reditus "". neco.

casus '☺'. opus reditus "☺". neco.

casus '☻'. opus reditus "☻". neco.

casus '♥'. opus reditus "♥". neco.

casus '♦'. opus reditus "♦". neco.

casus '♣'. opus reditus "♣". neco.

casus '♠'. opus reditus "♠". neco.

casus '•'. opus reditus "•". neco.

casus '◘'. opus reditus "◘". neco.

casus '○'. opus reditus "○". neco.

casus '◙'. opus reditus "◙". neco.

casus '♂'. opus reditus "♂". neco.

casus '♀'. opus reditus "♀". neco.

casus '♪'. opus reditus "♪". neco.

casus '♫'. opus reditus "♫". neco.

casus '☼'. opus reditus "☼". neco.

casus '►'. opus reditus "►". neco.

casus '◄'. opus reditus "◄". neco.

casus '↕'. opus reditus "↕". neco.

casus '‼'. opus reditus "‼". neco.

casus '¶'. opus reditus "¶". neco.

casus '§'. opus reditus "§". neco.

casus '▬'. opus reditus "▬". neco.

casus '↨'. opus reditus "↨". neco.

casus '↑'. opus reditus "↑". neco.

casus '↓'. opus reditus "↓". neco.

casus '→'. opus reditus "→". neco.

casus '←'. opus reditus "←". neco.

casus '∟'. opus reditus "∟". neco.

casus '↔'. opus reditus "↔". neco.

casus '▲'. opus reditus "▲". neco.

casus '▼'. opus reditus "▼". neco.

casus ' '. opus reditus " ". neco.

casus '!'. opus reditus "!". neco.

casus '"'. opus reditus """. neco.

casus '#'. opus reditus "#". neco.

casus '$'. opus reditus "$". neco.

casus '%'. opus reditus "%". neco.

casus '&'. opus reditus "&". neco.

casus '\''. opus reditus "'". neco.

casus '('. opus reditus "(". neco.

casus ')'. opus reditus ")". neco.

casus '\*'. opus reditus "\*". neco.

casus '+'. opus reditus "+". neco.

casus ','. opus reditus ",". neco.

casus '-'. opus reditus "-". neco.

casus '.'. opus reditus ".". neco.

casus '/'. opus reditus "/". neco.

casus '0'. opus reditus "0". neco.

casus '1'. opus reditus "1". neco.

casus '2'. opus reditus "2". neco.

casus '3'. opus reditus "3". neco.

casus '4'. opus reditus "4". neco.

casus '5'. opus reditus "5". neco.

casus '6'. opus reditus "6". neco.

casus '7'. opus reditus "7". neco.

casus '8'. opus reditus "8". neco.

casus '9'. opus reditus "9". neco.

casus ':'. opus reditus ":". neco.

casus ';'. opus reditus ";". neco.

casus '<'. opus reditus "<". neco.

casus '='. opus reditus "=". neco.

casus '>'. opus reditus ">". neco.

casus '?'. opus reditus "?". neco.

casus '@'. opus reditus "@". neco.

casus 'A'. opus reditus "A". neco.

casus 'B'. opus reditus "B". neco.

casus 'C'. opus reditus "C". neco.

casus 'D'. opus reditus "D". neco.

casus 'E'. opus reditus "E". neco.

casus 'F'. opus reditus "F". neco.

casus 'G'. opus reditus "G". neco.

casus 'H'. opus reditus "H". neco.

casus 'I'. opus reditus "I". neco.

casus 'J'. opus reditus "J". neco.

casus 'K'. opus reditus "K". neco.

casus 'L'. opus reditus "L". neco.

casus 'M'. opus reditus "M". neco.

casus 'N'. opus reditus "N". neco.

casus 'O'. opus reditus "O". neco.

casus 'P'. opus reditus "P". neco.

casus 'Q'. opus reditus "Q". neco.

casus 'R'. opus reditus "R". neco.

casus 'S'. opus reditus "S". neco.

casus 'T'. opus reditus "T". neco.

casus 'U'. opus reditus "U". neco.

casus 'V'. opus reditus "V". neco.

casus 'W'. opus reditus "W". neco.

casus 'X'. opus reditus "X". neco.

casus 'Y'. opus reditus "Y". neco.

casus 'Z'. opus reditus "Z". neco.

casus '['. opus reditus "[". neco.

casus '\'. opus reditus "\". neco.

casus ']'. opus reditus "]". neco.

casus '^'. opus reditus "^". neco.

casus '\_'. opus reditus "\_". neco.

casus '`'. opus reditus "`". neco.

casus 'a'. opus reditus "a". neco.

casus 'b'. opus reditus "b". neco.

casus 'c'. opus reditus "c". neco.

casus 'd'. opus reditus "d". neco.

casus 'e'. opus reditus "e". neco.

casus 'f'. opus reditus "f". neco.

casus 'g'. opus reditus "g". neco.

casus 'h'. opus reditus "h". neco.

casus 'i'. opus reditus "i". neco.

casus 'j'. opus reditus "j". neco.

casus 'k'. opus reditus "k". neco.

casus 'l'. opus reditus "l". neco.

casus 'm'. opus reditus "m". neco.

casus 'n'. opus reditus "n". neco.

casus 'o'. opus reditus "o". neco.

casus 'p'. opus reditus "p". neco.

casus 'q'. opus reditus "q". neco.

casus 'r'. opus reditus "r". neco.

casus 's'. opus reditus "s". neco.

casus 't'. opus reditus "t". neco.

casus 'u'. opus reditus "u". neco.

casus 'v'. opus reditus "v". neco.

casus 'w'. opus reditus "w". neco.

casus 'x'. opus reditus "x". neco.

casus 'y'. opus reditus "y". neco.

casus 'z'. opus reditus "z". neco.

casus '{'. opus reditus "{". neco.

casus '|'. opus reditus "|". neco.

casus '}'. opus reditus "}". neco.

casus 'Ï'. opus reditus "Ï". neco.

casus '⌂'. opus reditus "⌂". neco.

casus 'Ç'. opus reditus "Ç". neco.

casus 'ü'. opus reditus "ü". neco.

casus 'é'. opus reditus "é". neco.

casus 'â'. opus reditus "â". neco.

casus 'ä'. opus reditus "ä". neco.

casus 'à'. opus reditus "à". neco.

casus 'å'. opus reditus "å". neco.

casus 'ç'. opus reditus "ç". neco.

casus 'ê'. opus reditus "ê". neco.

casus 'ë'. opus reditus "ë". neco.

casus 'è'. opus reditus "è". neco.

casus 'ï'. opus reditus "ï". neco.

casus 'î'. opus reditus "î". neco.

casus 'ì'. opus reditus "ì". neco.

casus 'Ä'. opus reditus "Ä". neco.

casus 'Å'. opus reditus "Å". neco.

casus 'É'. opus reditus "É". neco.

casus 'æ'. opus reditus "æ". neco.

casus 'Æ'. opus reditus "Æ". neco.

casus 'ô'. opus reditus "ô". neco.

casus 'ö'. opus reditus "ö". neco.

casus 'ò'. opus reditus "ò". neco.

casus 'û'. opus reditus "û". neco.

casus 'ù'. opus reditus "ù". neco.

casus 'ÿ'. opus reditus "ÿ". neco.

casus 'Ö'. opus reditus "Ö". neco.

casus 'Ü'. opus reditus "Ü". neco.

casus 'ø'. opus reditus "ø". neco.

casus '£'. opus reditus "£". neco.

casus 'Ø'. opus reditus "Ø". neco.

casus '×'. opus reditus "×". neco.

casus 'ƒ'. opus reditus "ƒ". neco.

casus 'á'. opus reditus "á". neco.

casus 'í'. opus reditus "í". neco.

casus 'ó'. opus reditus "ó". neco.

casus 'ú'. opus reditus "ú". neco.

casus 'ñ'. opus reditus "ñ". neco.

casus 'Ñ'. opus reditus "Ñ". neco.

casus 'ª'. opus reditus "ª". neco.

casus 'º'. opus reditus "º". neco.

casus '¿'. opus reditus "¿". neco.

casus '®'. opus reditus "®". neco.

casus '¬'. opus reditus "¬". neco.

casus '½'. opus reditus "½". neco.

casus '¼'. opus reditus "¼". neco.

casus '¡'. opus reditus "¡". neco.

casus '«'. opus reditus "«". neco.

casus '»'. opus reditus "»". neco.

casus '░'. opus reditus "░". neco.

casus '▒'. opus reditus "▒". neco.

casus '▓'. opus reditus "▓". neco.

casus '│'. opus reditus "│". neco.

casus '┤'. opus reditus "┤". neco.

casus 'Á'. opus reditus "Á". neco.

casus 'Â'. opus reditus "Â". neco.

casus 'À'. opus reditus "À". neco.

casus '©'. opus reditus "©". neco.

casus '╣'. opus reditus "╣". neco.

casus '║'. opus reditus "║". neco.

casus '╗'. opus reditus "╗". neco.

casus '╝'. opus reditus "╝". neco.

casus '¢'. opus reditus "¢". neco.

casus '¥'. opus reditus "¥". neco.

casus '┐'. opus reditus "┐". neco.

casus '└'. opus reditus "└". neco.

casus '┴'. opus reditus "┴". neco.

casus '┬'. opus reditus "┬". neco.

casus '├'. opus reditus "├". neco.

casus '─'. opus reditus "─". neco.

casus '┼'. opus reditus "┼". neco.

casus 'ã'. opus reditus "ã". neco.

casus 'Ã'. opus reditus "Ã". neco.

casus '╚'. opus reditus "╚". neco.

casus '╔'. opus reditus "╔". neco.

casus '╩'. opus reditus "╩". neco.

casus '╦'. opus reditus "╦". neco.

casus '╠'. opus reditus "╠". neco.

casus '═'. opus reditus "═". neco.

casus '╬'. opus reditus "╬". neco.

casus '¤'. opus reditus "¤". neco.

casus 'ð'. opus reditus "ð". neco.

casus 'Ð'. opus reditus "Ð". neco.

casus 'Ê'. opus reditus "Ê". neco.

casus 'Ë'. opus reditus "Ë". neco.

casus 'È'. opus reditus "È". neco.

casus 'ı'. opus reditus "ı". neco.

casus 'Í'. opus reditus "Í". neco.

casus 'Î'. opus reditus "Î". neco.

casus 'Ï'. opus reditus "Ï". neco.

casus '┘'. opus reditus "┘". neco.

casus '┌'. opus reditus "┌". neco.

casus '█'. opus reditus "█". neco.

casus '▄'. opus reditus "▄". neco.

casus '¦'. opus reditus "¦". neco.

casus 'Ì'. opus reditus "Ì". neco.

casus '▀'. opus reditus "▀". neco.

casus 'Ó'. opus reditus "Ó". neco.

casus 'ß'. opus reditus "ß". neco.

casus 'Ô'. opus reditus "Ô". neco.

casus 'Ò'. opus reditus "Ò". neco.

casus 'õ'. opus reditus "õ". neco.

casus 'Õ'. opus reditus "Õ". neco.

casus 'µ'. opus reditus "µ". neco.

casus 'þ'. opus reditus "þ". neco.

casus 'Þ'. opus reditus "Þ". neco.

casus 'Ú'. opus reditus "Ú". neco.

casus 'Û'. opus reditus "Û". neco.

casus 'Ù'. opus reditus "Ù". neco.

casus 'ý'. opus reditus "ý". neco.

casus 'Ý'. opus reditus "Ý". neco.

casus '¯'. opus reditus "¯". neco.

casus '´'. opus reditus "´". neco.

casus ''. opus reditus "". neco.

casus '±'. opus reditus "±". neco.

casus '‗'. opus reditus "‗". neco.

casus 'À'. opus reditus "À". neco.

casus '¶'. opus reditus "¶". neco.

casus '§'. opus reditus "§". neco.

casus '÷'. opus reditus "÷". neco.

casus '¸'. opus reditus "¸". neco.

casus '°'. opus reditus "°". neco.

casus '¨'. opus reditus "¨". neco.

casus '·'. opus reditus "·". neco.

casus '¹'. opus reditus "¹". neco.

casus '³'. opus reditus "³". neco.

casus '²'. opus reditus "²". neco.

casus '■'. opus reditus "■". neco.

casus ' '. opus reditus " ". neco.

detrimentum reditus "". neco.

finis

.

finis

.

<# Pasa de imago a numerus #>

efficio imago\_to\_numerus(imago cht) : numerus

initum

reditus ascii\_imago\_to\_numerus(cht).

finis

.

<# Pasa de imago a catena #>

efficio imago\_to\_catena(imago cht) : catena.

initum

reditus ascii\_imago\_to\_catena(cht).

finis

.

<# Pasa de imago a dualis #>

efficio imago\_to\_dualis(imago cht) : dualis

initum

in cht = 'V' aut cht = 'v certus

reditus veridicus.

.

reditus falsidicus.

finis

.

<#--------------------------------------------

---------------Segmento de catena-------------

----------------------------------------------#>

<# Tabla de equivalencia acii para catena a imago#>

efficio ascii\_catena\_to\_imago(catena verb) : imago

initum

aeger verb

initum

casus "" opus reditus ''. neco.

casus "☺" opus reditus '☺'. neco.

casus "☻" opus reditus '☻'. neco.

casus "♥" opus reditus '♥'. neco.

casus "♦" opus reditus '♦'. neco.

casus "♣" opus reditus '♣'. neco.

casus "♠" opus reditus '♠'. neco.

casus "•" opus reditus '•'. neco.

casus "◘" opus reditus '◘'. neco.

casus "○" opus reditus '○'. neco.

casus "◙" opus reditus '◙'. neco.

casus "♂" opus reditus '♂'. neco.

casus "♀" opus reditus '♀'. neco.

casus "♪" opus reditus '♪'. neco.

casus "♫" opus reditus '♫'. neco.

casus "☼" opus reditus '☼'. neco.

casus "►" opus reditus '►'. neco.

casus "◄" opus reditus '◄'. neco.

casus "↕" opus reditus '↕'. neco.

casus "‼" opus reditus '‼'. neco.

casus "¶" opus reditus '¶'. neco.

casus "§" opus reditus '§'. neco.

casus "▬" opus reditus '▬'. neco.

casus "↨" opus reditus '↨'. neco.

casus "↑" opus reditus '↑'. neco.

casus "↓" opus reditus '↓'. neco.

casus "→" opus reditus '→'. neco.

casus "←" opus reditus '←'. neco.

casus "∟" opus reditus '∟'. neco.

casus "↔" opus reditus '↔'. neco.

casus "▲" opus reditus '▲'. neco.

casus "▼" opus reditus '▼'. neco.

casus " " opus reditus ' '. neco.

casus "!" opus reditus '!'. neco.

casus """ opus reditus '"'. neco.

casus "#" opus reditus '#'. neco.

casus "$" opus reditus '$'. neco.

casus "%" opus reditus '%'. neco.

casus "&" opus reditus '&'. neco.

casus "'" opus reditus '''. neco.

casus "(" opus reditus '('. neco.

casus ")" opus reditus ')'. neco.

casus "\*" opus reditus '\*'. neco.

casus "+" opus reditus '+'. neco.

casus "," opus reditus ','. neco.

casus "-" opus reditus '-'. neco.

casus "." opus reditus '.'. neco.

casus "/" opus reditus '/'. neco.

casus "0" opus reditus '0'. neco.

casus "1" opus reditus '1'. neco.

casus "2" opus reditus '2'. neco.

casus "3" opus reditus '3'. neco.

casus "4" opus reditus '4'. neco.

casus "5" opus reditus '5'. neco.

casus "6" opus reditus '6'. neco.

casus "7" opus reditus '7'. neco.

casus "8" opus reditus '8'. neco.

casus "9" opus reditus '9'. neco.

casus ":" opus reditus ':'. neco.

casus ";" opus reditus ';'. neco.

casus "<" opus reditus '<'. neco.

casus "=" opus reditus '='. neco.

casus ">" opus reditus '>'. neco.

casus "?" opus reditus '?'. neco.

casus "@" opus reditus '@'. neco.

casus "A" opus reditus 'A'. neco.

casus "B" opus reditus 'B'. neco.

casus "C" opus reditus 'C'. neco.

casus "D" opus reditus 'D'. neco.

casus "E" opus reditus 'E'. neco.

casus "F" opus reditus 'F'. neco.

casus "G" opus reditus 'G'. neco.

casus "H" opus reditus 'H'. neco.

casus "I" opus reditus 'I'. neco.

casus "J" opus reditus 'J'. neco.

casus "K" opus reditus 'K'. neco.

casus "L" opus reditus 'L'. neco.

casus "M" opus reditus 'M'. neco.

casus "N" opus reditus 'N'. neco.

casus "O" opus reditus 'O'. neco.

casus "P" opus reditus 'P'. neco.

casus "Q" opus reditus 'Q'. neco.

casus "R" opus reditus 'R'. neco.

casus "S" opus reditus 'S'. neco.

casus "T" opus reditus 'T'. neco.

casus "U" opus reditus 'U'. neco.

casus "V" opus reditus 'V'. neco.

casus "W" opus reditus 'W'. neco.

casus "X" opus reditus 'X'. neco.

casus "Y" opus reditus 'Y'. neco.

casus "Z" opus reditus 'Z'. neco.

casus "[" opus reditus '['. neco.

casus "\" opus reditus '\'. neco.

casus "]" opus reditus ']'. neco.

casus "^" opus reditus '^'. neco.

casus "\_" opus reditus '\_'. neco.

casus "`" opus reditus '`'. neco.

casus "a" opus reditus 'a'. neco.

casus "b" opus reditus 'b'. neco.

casus "c" opus reditus 'c'. neco.

casus "d" opus reditus 'd'. neco.

casus "e" opus reditus 'e'. neco.

casus "f" opus reditus 'f'. neco.

casus "g" opus reditus 'g'. neco.

casus "h" opus reditus 'h'. neco.

casus "i" opus reditus 'i'. neco.

casus "j" opus reditus 'j'. neco.

casus "k" opus reditus 'k'. neco.

casus "l" opus reditus 'l'. neco.

casus "m" opus reditus 'm'. neco.

casus "n" opus reditus 'n'. neco.

casus "o" opus reditus 'o'. neco.

casus "p" opus reditus 'p'. neco.

casus "q" opus reditus 'q'. neco.

casus "r" opus reditus 'r'. neco.

casus "s" opus reditus 's'. neco.

casus "t" opus reditus 't'. neco.

casus "u" opus reditus 'u'. neco.

casus "v" opus reditus 'v'. neco.

casus "w" opus reditus 'w'. neco.

casus "x" opus reditus 'x'. neco.

casus "y" opus reditus 'y'. neco.

casus "z" opus reditus 'z'. neco.

casus "{" opus reditus '{'. neco.

casus "|" opus reditus '|'. neco.

casus "}" opus reditus '}'. neco.

casus "Ï" opus reditus 'Ï'. neco.

casus "⌂" opus reditus '⌂'. neco.

casus "Ç" opus reditus 'Ç'. neco.

casus "ü" opus reditus 'ü'. neco.

casus "é" opus reditus 'é'. neco.

casus "â" opus reditus 'â'. neco.

casus "ä" opus reditus 'ä'. neco.

casus "à" opus reditus 'à'. neco.

casus "å" opus reditus 'å'. neco.

casus "ç" opus reditus 'ç'. neco.

casus "ê" opus reditus 'ê'. neco.

casus "ë" opus reditus 'ë'. neco.

casus "è" opus reditus 'è'. neco.

casus "ï" opus reditus 'ï'. neco.

casus "î" opus reditus 'î'. neco.

casus "ì" opus reditus 'ì'. neco.

casus "Ä" opus reditus 'Ä'. neco.

casus "Å" opus reditus 'Å'. neco.

casus "É" opus reditus 'É'. neco.

casus "æ" opus reditus 'æ'. neco.

casus "Æ" opus reditus 'Æ'. neco.

casus "ô" opus reditus 'ô'. neco.

casus "ö" opus reditus 'ö'. neco.

casus "ò" opus reditus 'ò'. neco.

casus "û" opus reditus 'û'. neco.

casus "ù" opus reditus 'ù'. neco.

casus "ÿ" opus reditus 'ÿ'. neco.

casus "Ö" opus reditus 'Ö'. neco.

casus "Ü" opus reditus 'Ü'. neco.

casus "ø" opus reditus 'ø'. neco.

casus "£" opus reditus '£'. neco.

casus "Ø" opus reditus 'Ø'. neco.

casus "×" opus reditus '×'. neco.

casus "ƒ" opus reditus 'ƒ'. neco.

casus "á" opus reditus 'á'. neco.

casus "í" opus reditus 'í'. neco.

casus "ó" opus reditus 'ó'. neco.

casus "ú" opus reditus 'ú'. neco.

casus "ñ" opus reditus 'ñ'. neco.

casus "Ñ" opus reditus 'Ñ'. neco.

casus "ª" opus reditus 'ª'. neco.

casus "º" opus reditus 'º'. neco.

casus "¿" opus reditus '¿'. neco.

casus "®" opus reditus '®'. neco.

casus "¬" opus reditus '¬'. neco.

casus "½" opus reditus '½'. neco.

casus "¼" opus reditus '¼'. neco.

casus "¡" opus reditus '¡'. neco.

casus "«" opus reditus '«'. neco.

casus "»" opus reditus '»'. neco.

casus "░" opus reditus '░'. neco.

casus "▒" opus reditus '▒'. neco.

casus "▓" opus reditus '▓'. neco.

casus "│" opus reditus '│'. neco.

casus "┤" opus reditus '┤'. neco.

casus "Á" opus reditus 'Á'. neco.

casus "Â" opus reditus 'Â'. neco.

casus "À" opus reditus 'À'. neco.

casus "©" opus reditus '©'. neco.

casus "╣" opus reditus '╣'. neco.

casus "║" opus reditus '║'. neco.

casus "╗" opus reditus '╗'. neco.

casus "╝" opus reditus '╝'. neco.

casus "¢" opus reditus '¢'. neco.

casus "¥" opus reditus '¥'. neco.

casus "┐" opus reditus '┐'. neco.

casus "└" opus reditus '└'. neco.

casus "┴" opus reditus '┴'. neco.

casus "┬" opus reditus '┬'. neco.

casus "├" opus reditus '├'. neco.

casus "─" opus reditus '─'. neco.

casus "┼" opus reditus '┼'. neco.

casus "ã" opus reditus 'ã'. neco.

casus "Ã" opus reditus 'Ã'. neco.

casus "╚" opus reditus '╚'. neco.

casus "╔" opus reditus '╔'. neco.

casus "╩" opus reditus '╩'. neco.

casus "╦" opus reditus '╦'. neco.

casus "╠" opus reditus '╠'. neco.

casus "═" opus reditus '═'. neco.

casus "╬" opus reditus '╬'. neco.

casus "¤" opus reditus '¤'. neco.

casus "ð" opus reditus 'ð'. neco.

casus "Ð" opus reditus 'Ð'. neco.

casus "Ê" opus reditus 'Ê'. neco.

casus "Ë" opus reditus 'Ë'. neco.

casus "È" opus reditus 'È'. neco.

casus "ı" opus reditus 'ı'. neco.

casus "Í" opus reditus 'Í'. neco.

casus "Î" opus reditus 'Î'. neco.

casus "Ï" opus reditus 'Ï'. neco.

casus "┘" opus reditus '┘'. neco.

casus "┌" opus reditus '┌'. neco.

casus "█" opus reditus '█'. neco.

casus "▄" opus reditus '▄'. neco.

casus "¦" opus reditus '¦'. neco.

casus "Ì" opus reditus 'Ì'. neco.

casus "▀" opus reditus '▀'. neco.

casus "Ó" opus reditus 'Ó'. neco.

casus "ß" opus reditus 'ß'. neco.

casus "Ô" opus reditus 'Ô'. neco.

casus "Ò" opus reditus 'Ò'. neco.

casus "õ" opus reditus 'õ'. neco.

casus "Õ" opus reditus 'Õ'. neco.

casus "µ" opus reditus 'µ'. neco.

casus "þ" opus reditus 'þ'. neco.

casus "Þ" opus reditus 'Þ'. neco.

casus "Ú" opus reditus 'Ú'. neco.

casus "Û" opus reditus 'Û'. neco.

casus "Ù" opus reditus 'Ù'. neco.

casus "ý" opus reditus 'ý'. neco.

casus "Ý" opus reditus 'Ý'. neco.

casus "¯" opus reditus '¯'. neco.

casus "´" opus reditus '´'. neco.

casus "" opus reditus ''. neco.

casus "±" opus reditus '±'. neco.

casus "‗" opus reditus '‗'. neco.

casus "À" opus reditus 'À'. neco.

casus "¶" opus reditus '¶'. neco.

casus "§" opus reditus '§'. neco.

casus "÷" opus reditus '÷'. neco.

casus "¸" opus reditus '¸'. neco.

casus "°" opus reditus '°'. neco.

casus "¨" opus reditus '¨'. neco.

casus "·" opus reditus '·'. neco.

casus "¹" opus reditus '¹'. neco.

casus "³" opus reditus '³'. neco.

casus "²" opus reditus '²'. neco.

casus "■" opus reditus '■'. neco.

casus " " opus reditus ' '. neco.

detrimentum reditus ''. neco.

finis

.

finis

.

<# Pasa de catena a imago #>

efficio catena\_to\_imago(catena verb) : imago

commutabilis

numerus total est 0.

numerus count est 0.

numerus long est verb$#.

catena temp\_cat est "".

imago temp\_imago est ''.

initum

tempus long > 0 opus

initum

temp\_cat := verb $1|<1.

in long > 1 certus

verb := verb $2|<long.

.

temp\_imago := ascii\_catena\_to\_imago(temp\_cat).

total += ascii\_imago\_to\_numerus(temp\_imago).

incrementum count.

decrementum long.

finis

.

reditus ascii\_numerus\_to\_imago(total/count).

finis

.

<# Tabla de caracteres posbiles dentro de un hexadecimal#>

efficio suus\_hex\_aux(imago cht): numerus

efficio

aeger cht

initum

casus '0' opus reditus 0. neco.

casus '1' opus reditus 1. neco.

casus '2' opus reditus 2. neco.

casus '3' opus reditus 3. neco.

casus '4' opus reditus 4. neco.

casus '5' opus reditus 5. neco.

casus '6' opus reditus 6. neco.

casus '7' opus reditus 7. neco.

casus '8' opus reditus 8. neco.

casus '9' opus reditus 9. neco.

casus 'a' opus reditus 10. neco.

casus 'b' opus reditus 11. neco.

casus 'c' opus reditus 12. neco.

casus 'd' opus reditus 13. neco.

casus 'e' opus reditus 14. neco.

casus 'f' opus reditus 15. neco.

detrimentum reditus  -1. neco.

finis

.

finis

.

<# Verifica que la catena suministrada sea un hexadecimal valido#>

efficio suus\_hex(catena verb): dualis

commutabili

numerus long est verb$#.

catena temp\_cat est "".

imago temp\_imago est ''.

initum

tempus long > 0 opus

initum

temp\_cat := verb $1|<1.

in long > 1 certus

verb := verb $2|<long.

.

temp\_imago := ascii\_catena\_to\_imago(temp\_cat).

temp\_imago[<<].

in suus\_hex\_aux(temp\_imago) = -1 certus

reditus falsidicus.

.

decrementum long.

finis

.

reditus veridicus.

finis

.

<# Tabla de caracteres posibles dentro de un numero romano#>

efficio suus\_rom\_aux(imago cht): numerus

efficio

aeger cht

initum

casus 'i' opus reditus 1. neco.

casus 'v' opus reditus 5. neco.

casus 'x' opus reditus 10. neco.

casus 'l' opus reditus 50. neco.

casus 'c' opus reditus 100. neco.

casus 'd' opus reditus 500. neco.

casus 'm' opus reditus 1000. neco.

detrimentum reditus -1 . neco.

finis

.

finis

.

<# Verifica que la catena suministrada sea un numero romano valido#>

efficio suus\_rom(catena verb): dualis

commutabilis

numerus long est verb$#.

catena temp\_cat est "".

imago temp\_imago est ''.

initum

tempus long > 0 opus

initum

temp\_cat := verb $1|<1.

in long > 1 certus

verb := verb $2|<long.

.

temp\_imago := ascii\_catena\_to\_imago(temp\_cat).

temp\_imago[<<].

in suus\_rom\_aux(temp\_imago) = -1 certus

reditus falsidicus.

.

decrementum long.

finis

.

reditus veridicus.

finis

.

<# Tabla de caracteres posibles dentro de un numero entero#>

efficio suus\_num\_aux(imago cht): numerus

efficio

aeger cht

initum

casus '0' opus reditus 0. neco.

casus '1' opus reditus 1. neco.

casus '2' opus reditus 2. neco.

casus '3' opus reditus 3. neco.

casus '4' opus reditus 4. neco.

casus '5' opus reditus 5. neco.

casus '6' opus reditus 6. neco.

casus '7' opus reditus 7. neco.

casus '8' opus reditus 8. neco.

casus '9' opus reditus 9. neco.

detrimentum reditus -1. neco.

finis

.

finis

.

<# Verifica que la catena suministrada sea un numero valido#>

efficio suus\_num(catena verb): dualis commutabilis

numerus long est verb$#.

catena temp\_cat est "".

imago temp\_imago est ''.

initum

in verb $1|<1 = "-" certus

in long > 1 certus

verb := verb $2|<long.

.

.

tempus long > 0 opus

initum

temp\_cat := verb $1|<1.

in long > 1 certus

verb := verb $2|<long.

.

temp\_imago := ascii\_catena\_to\_imago(temp\_cat).

temp\_imago[<<].

in suus\_num\_aux(temp\_imago) -1 certus

reditus falsidicus.

.

decrementum long.

finis

.

reditus veridicus.

finis

.

<# Pasa un número hexadecimal dado por una catena a entero #>

efficio catena\_hex\_to\_numerus(catena verb): numerus

commutabilis

numerus total est 0.

numerus temp\_val est 0.

numerus long est verb$#.

efficio

panis I := long-1 auctum -1 gradus -1 opus

temp\_val := suus\_hex\_aux(verb $i|<i).

total = total \*16 + temp\_val.

.

reditus total.

finis

.

<#  Pasa un número entero dado por una catena a entero #>

efficio catena\_rom\_to\_numerus(catena verb): numerus

commutabilis

numerus total est 0.

numerus actual\_sym\_value est 0.

numerus next\_sym\_value est 0.

numerus long est verb$#.

efficio

panis I := long-1 auctum -1 gradus -1 opus

actual\_sym\_value := suus\_rom\_aux(verb $i|<i).

in i > 0

certus

next\_sym\_value := suus\_rom\_aux(verb $i-1|<i-1).

in next\_sym\_value >= actual\_sym\_value

certus

total -= actual\_sym\_value.

mentiri

total += actual\_sym\_value.

.

mentiri

total += actual\_sym\_value.

.

.

reditus total.

finis

.

<# Funsión de potencia #>

efficio potencia(numerus base, numerus exponente): numerus

commutabilis

numerus total est 1.

efficio

panis i := 0 auctum exponente gradus 1 opus

total := total \* base

.

finis

.

<#  Pasa un número romano dado por una catena a entero #>

efficio catena\_num\_to\_numerus(catena verb): numerus

commutabilis

numerus total est 0.

numerus long est verb$#.

catena temp\_cat est "".

imago temp\_imago est ''.

dualis vexillum est falsidicus.

efficio

in (verb $1|<1) = "-" certus

vexillum =: veridicus.

verb := verb $2|<long.

.

panis I := 0 auctum long gradus 1 opus

temp\_cat := verb $i|<i.

verb := verb $i|<long.

temp\_imago := ascii\_catena\_to\_imago(temp\_cat).

total := total + (suus\_num\_aux(temp\_imago) \* (potencia(10, i)))

.

in vexillum certus

total := total \* (-1).

.

reditus total.

finis

.

<# Verifica si la catena es un entero, un hexadecimal o un romano y procede a convertilo a numerus #>

efficio numerus\_hex\_rom(catena verb): catena

initum

in (verb $1|<2) = "0X" aut (verb $1|<2) = "0x" certus

verb := (verb $2|<verb$# - 1)

in suus\_hex(verb) certus

reditus catena\_hex\_to\_numerus(verb).

.

reditus 0

.

in (verb $1|<2) = "0R" aut (verb $1|<2) = "0r" certus

verb := (verb $2|<verb$# - 1)

in suus\_rom(verb) certus

reditus catena\_rom\_to\_numerus(verb).

.

reditus 0

.

in suus\_num(verb) certus

reditus catena\_num\_to\_numerus(verb).

.

reditus 0

finis

.

<# Pasa de catena a numerus #>

efficio catena\_to\_numerus(catena verb) : numerus

initum

reditus numerus\_hex\_rom(verb).

finis

.

<# Pasa de catena a dualis #>

efficio catena\_to\_dualis(catena verb) : dualis

commutabilis

numerus long est verb$#.

catena temp\_cat est "".

imago temp\_imago est ''.

catena final\_cat est "".

initum

tempus long > 0 opus

initum

temp\_cat := verb $1|<1.

in long > 1 certus

verb := verb $2|<long.

.

temp\_imago := ascii\_catena\_to\_imago(temp\_cat).

temp\_imago[<<].

final\_cat = final\_cat $+ ascii\_imago\_to\_catena(temp\_imago).

decrementum long.

finis

.

in final\_cat = "veridicus"

certus

reditus veridicus.

mentiri

reditus falsidicus.

.

finis

.

<#--------------------------------------------

---------------Segmento de dualis-------------

----------------------------------------------#>

<# Pasa de dualis a numerus #>

efficio dualis\_to\_dualis(dualis N) : numerus

initum

in veridicus

certus

reditus 1.

mentiri

reditus 0.

.

finis

.

<# Pasa de dualis a imago #>

efficio dualis\_to\_imago(dualis N) : imago.

initum

in veridicus

certus

reditus 'V'.

mentiri

reditus 'F'.

.

finis

.

<# Pasa de dualis a catena #> falta

efficio dualis\_to\_catena(dualis N) : catena

in veridicus

certus

reditus "veridicus".

mentiri

reditus "falsidicus".

.

finis

.