

Versión 1.0

2020

NumerosAstronomicos.h



UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Autores:
Nicolás Carson
Ramiro Flores

ÍNDICE

MANUAL DE USUARIO	3
Módulo NumeroAstronomico.h	3
Módulo errorHandler.h	6
Módulo astroFile.h	8
Módulo arithmetic.h	10
Pruebas unitarias	12
Implementación	12
Demo	13
Conclusión	13

INTRODUCCIÓN

En esta introducción explicaremos un poco como fue el proceso, y el camino que recorrimos en la creación de este código en lenguaje C. Para comenzar intentaremos ponerlos en sintonía explicando a grandes rasgos cuáles eran las metas a alcanzar, estas metas eran crear un batería de funciones que puedan manejar un nuevo tipo de dato struct llamado "Número astronómico", ¿Qué es esto?, fácil un tipo de dato formado por dos partes un puntero a una cadena de caracteres que contendrá los caracteres que forman este número con una longitud máxima de 100, y su otra parte que es un entero que contendrá o bien la longitud del número o bien un entero que represente un error si este lo tuviera.

Una vez planteada la meta nuestro primer desafío fue cómo iba a estar representado este número dentro de la cadena, si su dígito más importante iba a estar por delante o por detrás de la cadena o si se dejaba o no un lugar para las flags, estas decisiones se tomaron analizando cómo convendría tener la cadena a la hora de pasar estos datos o como sería más fácil cargar estos datos. Una vez resuelto el primer obstáculo la siguiente tarea fue organizar cuales iban a ser las funciones convendría programarlas primero, esta necesidad surge de que como algunas de estas usan directa o indirectamente las otras, una buena organización antes de lanzarnos fue importante. Ya resuelta la organización previa ya pudimos ir directo a empezar con la escritura del código, donde fuimos completando los distintos módulos o grupos de funciones que tenían una cierta relación entre sí en el orden antes definido, y aunque no fue fácil y más de una vez hubo que chocar contra una pared para volver atrás y mejorar nuestra visión del panorama, se llegó a estas metas antes planteadas y esto gracias a una buena organización previa, una buena sociedad y la voluntad de seguir aprendiendo cada vez que se ignoraba o se tenía una visión tanto errada como poco óptima de cara a lo que es el lenguaje C.

MANUAL DE USUARIO

La biblioteca fue modularizada para mayor escalabilidad y organización del código

Módulo NumeroAstronomico.h

char *obtenerSoloValor(NumeroAstronomico nro)

Recibe un número al cual se le elimina las banderas y los ceros

Parámetros

Nombre	Tipo	Descripción
nro	NumeroAstronomico	Todo número astronómico tiene su parte entera

char *obtenerBanderas(NumeroAstronomico nro)

Recibe un NA y devuelve sus banderas

Parámetros

Nombre	Tipo	Descripción
nro	NumeroAstronomico	Todo número astronómico tiene sus banderas (carry y overflow)

NumeroAstronomico crearDesdeCadena(char *cadena)

Recibe una cadena de caracteres y si es válida genera un NA con el contenido de esta

Parámetros

Nombre	Tipo	Descripción
cadena	char*	Es necesario una cadena con números

NumeroAstronomico crearDesdeCifraSeguidaDeCeros(int cifra, int cantCeros)

Recibe una cadena de caracteres y si es válida genera un NA con el contenido de esta

Parámetros

Nombre	Tipo	Descripción
cifra	int	Cifras para el número astronómico a crear
cantCeros	int	Cantidad de ceros para el número astronómico a crear

NumeroAstronomico crearAleatorio(void)

Genera un NA aleatorio tanto es su longitud(100 como máximo) como en su contenido

Parámetros

Nombre	Tipo	Descripción
(No tiene)		

FILE *mostrar(NumeroAstronomico nro, int grupoEnPrimerLinea, FILE *flujo)

Recibe un número, una cantidad de grupos en la primera línea y un flujo igualado a stdout por lo que apunta a la pantalla de la consola. Su función es mostrar el número en consola de forma tal que aparecen en la primer línea la cantidad dada de grupos de 3 números (salvo el primero que puede ser de 3, 2 o 1) separados por un punto, en las siguientes líneas se muestran una cantidad de 1 menos que la primera de grupos por línea (siempre que la longitud lo permita), y termina con una línea con la cantidad de grupos sobrantes si estos existiesen.

Parámetros

Nombre	Tipo	Descripción
nro	NumeroAstronomico	Es un número astronómico
grupoEnPrimerLinea	int	Cantidad de grupos en primer línea
flujo	FILE*	Flujo de un archivo

const char *mostrarLinea(FILE *flujo, const char *ptr, int longitud)

Recibe un flujo igualado a stdout, un puntero a char y una longitud, y muestra una línea la cantidad de números dada por la longitud comenzando por donde apunte el puntero, estos están separados por un punto en grupos de 3 (salvo el primero que puede ser 3, 2 o 1)

Parámetros

Nombre	Tipo	Descripción
flujo	FILE*	Flujo de un archivo
ptr	char*	Un puntero a un carácter
longitud	int	Una longitud

Módulo errorHandler.h

int esSecuenciaNula(NumeroAstronomico nro)

A partir de un NA se fija si este es nulo o existe y devuelve TRUE(1) si existiese o FALSE(0) si no

Parámetros

Nombre	Tipo	Descripción
nro	NumeroAstronomico	Es un número astronómico

int esOverflow(NumeroAstronomico nro)

A partir de un NA se fija si este excede la longitud maxima de caracteres(100) y devuelve TRUE(1) si la excede o FALSE(0) si no

Parámetros

Nombre	Tipo	Descripción
nro	NumeroAstronomico	Es un número astronómico

int esSecuenciaInvalida(NumeroAstronomico nro)

A partir de un NA se fija si este contiene un caracter invalido(cualquiera que no fuese un digito) y devuelve TRUE(1) si lo contiene o FALSE(0) si no

Parámetros

Nombre	Tipo	Descripción
nro	NumeroAstronomico	Es un número astronómico

int esError(**NumeroAstronomico** nro)

A partir de un NA se fija si este tiene algún tipo de error(secuencia nula, secuencia invalida o overflow) y devuelve TRUE(1) si lo tiene o FALSE(0) si no

Parámetros

Nombre	Tipo	Descripción
nro	NumeroAstronomico	Es un número astronómico

int getTipoDeError(**NumeroAstronomico** nro)

A partir de un NA se fija si este contiene algún error y devuelve SECUENCIA_NULA(-1), SECUENCIA_INVALIDA(-2), OVER_FLOW(-3), DOBLE_ERROR(-4)(este se produce cuando hay tanto secuencia invalida como overflow) o FALSE(0) si no se encontrase error

Parámetros

Nombre	Tipo	Descripción
nro	NumeroAstronomico	Es un número astronómico

void printError(**NumeroAstronomico** numero)

A partir de un NA se fija que tipo de error tiene y muestra por pantalla un mensaje de error con el error correspondiente

Parámetros

Nombre	Tipo	Descripción
numero	NumeroAstronomico	Es un número astronómico

Módulo astroFile.h

void show(NumeroAstronomico nro)

No devuelve nada pero muestra por pantalla el número astronómico de manera legible

Parámetros

Nombre	Tipo	Descripción
nro	NumeroAstronomico	Es un número astronómico

NumeroAstronomico readNA(FILE *flujo)

Genera un NA a partir de un puntero a un archivo, este debe contener el registro de un NA

Parámetros

Nombre	Tipo	Descripción
flujo	FILE *	Flujo de un archivo

void writeNA(NumeroAstronomico nroAstronomico, FILE *flujo)

A partir de un NA y un puntero a un archivo, escribe el NA en un archivo binario

Parámetros

Nombre	Tipo	Descripción
nroAstronomico	NumeroAstronomico	Es un número astronómico
flujo	FILE *	Flujo de un archivo

NumeroAstronomico scanNA(FILE *flujo)

Genera un NA apartir de un puntero a un archivo, este debe contener el nuevo numero con el formato: *secuenciadedigitos#*

Parámetros

Nombre	Tipo	Descripción
flujo	FILE *	Flujo de un archivo

void printNA(NumeroAstronomico nro, FILE *flujo)

A partir de un NA y un puntero a un archivo, escribe el NA dado en el archivo apuntado con el formato: *secuenciadedigitos#*

Parámetros

Nombre	Tipo	Descripción
nroAstronomico	NumeroAstronomico	Es un número astronómico
flujo	FILE *	Flujo de un archivo

Módulo arithmetic.h

int esMenor(NumeroAstronomico primerNro, NumeroAstronomico segundoNro)

Se evalua si primerNro es menor al segundoNro y devuelve TRUE(1) si lo es o FALSE(0) si no

Parámetros

Nombre	Tipo	Descripción
primerNro	NumeroAstronomico	Es un número astronómico
segundoNro	NumeroAstronomico	Es un número astronómico

int sonIguales(NumeroAstronomico primerNro, NumeroAstronomico segundoNro)

A partir de dos NAs se fija si estos son iguales y devuelve TRUE(1) si lo son o FALSE(0) si no.

Además se determina el valor de verdad si primerNro y segundoNro si tienen igualdad en cuanto longitud y si los dígitos coincide

Parámetros

Nombre	Tipo	Descripción
primerNro	NumeroAstronomico	Es un número astronómico
segundoNro	NumeroAstronomico	Es un número astronómico

NumeroAstronomico sumar(**NumeroAstronomico** primerNro,
NumeroAstronomico segundoNro)

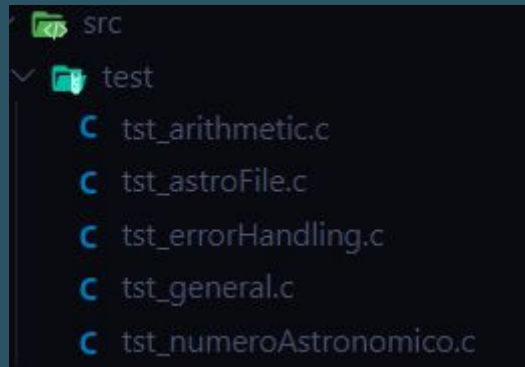
Recibe dos números y realiza suma digito a digito utilizando banderas

Parámetros

Nombre	Tipo	Descripción
primerNro	NumeroAstronomico	Es un número astronómico
segundoNro	NumeroAstronomico	Es un número astronómico

Pruebas unitarias

Se pueden observar ejemplos de implementación dentro del código fuente en la carpeta "test"



Implementación

Para poder empezar a utilizar la biblioteca bastará con importar los módulos que se desea

```
#include <stdio.h>
#include "../src/arithmetic.h"
#include "../src/numeroAstronomico.h"
#include "../src/astroFile.h"
#include "../src/errorHandling.h"

int main()
{
    NumeroAstronomico a = crearAleatorio();
}
```

Demo

Puede observar un ejemplo de implementación

```
raflores:~/Desktop/numerosAstro(dev)$ gcc demo.c && ./a.out
Bienvenidos a la biblioteca NumerosAstronomicos.h, presione...
(1) Para saber si un numero es menor que otro.
(2) Para saber si dos numeros son iguales.
(3) Para sumar dos numeros.
(4) Mostrar numero.
1
Presione...
(1) Crear un numero astronomico a partir de cadena
(2) Crear un numero astronomico aleatorio
(3) Crear un numero astronomico desde cifras seguidas de cero
1

Escriba la cadena que desea crear
200

ingrese otro numero
Presione...
(1) Crear un numero astronomico a partir de cadena
(2) Crear un numero astronomico aleatorio
(3) Crear un numero astronomico desde cifras seguidas de cero
1
```

Conclusión

A partir del diseño del software brindada por la facultad UTN FRBA pudimos realizar una implementación en ANSI C haciendo una construcción de bibliotecas permitiendo la modularización, el desacoplamiento y la centralización de funcionalidades comunes.

Puede acceder a nuestro proyecto a través del siguiente enlace:

<https://github.com/ramirosacruz/numerosAstro/>