

Bloque 1: programación secuencial.

Práctica 1: Series numéricas y cambio de bases.

Curso programación.

mauriciodg00@gmail.com

*Por: Mauricio de Garay Hernández (con base en la práctica diseñada por Joel Romero Gómez).
Formato IEEE.*

Abstract—

En esta práctica realizarán un programa en el que haya un menú con dos opciones: obtener todas las series numéricas de 3 o más números consecutivos que sumados te den el valor ingresado por el usuario; o cambiar un valor ingresado a 3 bases (de base 2 a 16) que diga el usuario.

I. Objetivos

- ☐ Practicar ciclos y condiciones anidadas.
- ☐ Poder crear menús y programas cíclicos que no terminen hasta que el usuario lo desee.
- ☐ Reforzar su conocimiento de pointers y paso de argumentos.
- ☐ Reforzar sus buenas prácticas de programación: reutilización de código, paso de argumentos adecuado, returns, comentarios, etc.

II. Descripción del sistema.

Se desea un programa que tenga la capacidad de llevar a cabo tres funciones manejadas a través de un menú:

1. Recibir un número entero positivo del usuario, encontrar todas las series de 3 o más números consecutivos que sumados den como resultado este valor.
2. Recibir un número entero positivo del usuario, convertirlo a 3 bases (de igual manera leídas del usuario) entre 2 y 16.
3. Salir del programa

Además, antes de desplegar el menú, se debe de enseñar el nombre del creador, título del programa y fecha.

III. Alcances y limitaciones

1. Todos los valores deben de ingresarse a través del teclado a una terminal.
2. Se deben de validar que todos los números sean enteros positivos.
Además, la opción del menú debe de ser 1, 2 o 3 (validar que no se ingrese una opción inexistente).

3. Para bases 11 a 16, los dígitos mayores a 9 se deben de expresar como letras (Ejemplo: 10 en hexadecimal es A, hexadecimal tiene dígitos de 0 a F, base 15 de 0 a E, etc).

IV. Requisitos funcionales.

El programa debe de subirse a su rama de github en el apartado Bloque1/Prácticas/Practical1.

Aquí, debe de estar la documentación más el código .c que realizaron. La documentación debe de tener: especificación de requerimientos (con entradas, salidas, alcances, limitaciones, diseños de pantalla, etc), diagramas de diseño (a su preferencia) y un pseudocódigo **muy** sencillo (no tiene que estar completamente detallado).

Al comenzar, el programa debe de desplegar al usuario el nombre del desarrollador y la fecha en que se creó, esperando a que el usuario ingrese <enter> (lean una variable tipo char “basura” y posteriormente hagan fflush(stdin)) para continuar. Una vez que el usuario ingrese <enter>, se limpiará la pantalla (system(“clear”) en mac/linux o system(“cls”) en windows).

A continuación, se desplegará un menú con 3 opciones: series numéricas, cambio de bases y salir. Se deberá de validar que no se ingrese una opción diferente a estas 3. Además, una vez que se despliegan los resultados de la opción 1 o 2, se regresará al menú, así permitiendo que el programa sea cíclico y no termine hasta que el usuario ingrese la opción 3.

Si se escoge la opción 1, se pedirá un número **entero y positivo** (se debe validar), y posteriormente se demostrarán todas las series de 3 o más números consecutivos que tengan como resultado este valor ingresado. Por ejemplo:

Dame un número entero: 156

6 + 7 + 8 + 9 + 10 + 11 + 12 + 13 + 14 + 15 + 16 + 17 + 18 = 156

16 + 17 + 18 + 19 + 20 + 21 + 22 + 23 = 156

51 + 52 + 53 = 156

Terminando, se devuelve al menú, donde se esperará otra opción.

Para la segunda opción, se le pedirá al usuario que ingrese un número **entero y positivo** (se deberá validar), y 3 bases de la forma: base1, base2, base3 (validen el formato con los valores de return de scanf()). El resultado será la conversión de este número a cada una de las bases numéricas. Por ejemplo:

Dame un número entero: 44

Dame 3 bases numéricas

separadas por comas y espacios

(base1, base2, base3): 2, 14, 15

El número en base 2 es: 101100.

El número en base 16 es: 2C.

El número en base 15 es: 2E.

Terminando, se devuelve al menú, donde se esperará otra opción.

Si se ingresa la opción 3, se le darán las gracias al usuario y terminará la ejecución del programa.

V. Restricciones de programación

- No se pueden utilizar arreglos (ni por notación de corchetes o como pointers) ni matrices.
- No se puede utilizar recursión (que una función se llame a sí misma).
- No se puede utilizar switch-case.
- No se pueden utilizar estructuras/crear tipos de datos.
- No se pueden utilizar variables globales.
- No se puede utilizar fuerza bruta (usen ciclos).

- Utilizar funciones.
- Utilizar aunque sea un return.
- Utilizar aunque sea un define.
- Hacer comentarios por funciones y por archivo como se vio en clase.
- No se puede utilizar alguna librería distinta a: stdio.h, stdlib.h, math.h.

VI.- Fecha de entrega

Miércoles 30 de diciembre del 2020, a través de github.