TALLER PRÁCTICO

PROGRAMACIÓN 2

PRF: OMAR IVAN TREJOS B

ING(C). GABRIEL GUTIERREZ T.
T.I 9801196066

ING(C). HÉCTOR FABIO JIMÉNEZ S.
C.C 1.115.421.345

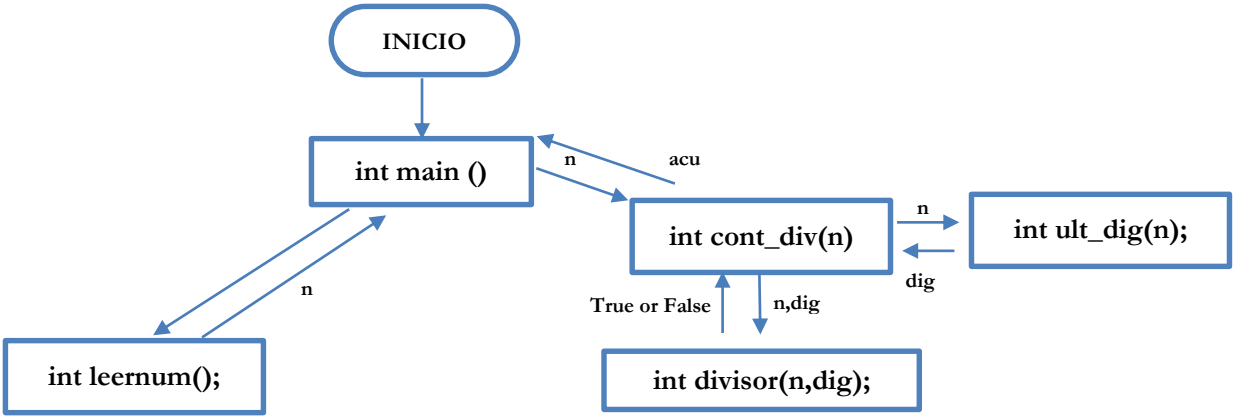
INGENIERIA DE SISTEMAS Y COMPUTACIÓN
UNIVERSIDAD TECNOLOGICA DE PEREIRA
2016-1

SOLUCIÓN

GABRIEL GUTIERREZ T, HÉCTOR F. JIMÉNEZ S.
ggutierreztamayo@utp.edu.co, hfjimenez@utp.edu.co[1]

1. Leer un número ***Z*** y determinar ***cuántos de sus dígitos*** lo divide exactamente.

I. DIAGRAMA FUNCIONAL



II. OBJETIVO DE FUNCIONES

- int leernum()[**Interface**]: Es una interface para obtener el numero **Z** que ingresa el usuario.
- int cont_div(int n)[**Operativa**]: Se encarga de contar los divisores desde n hasta 0, cuando termina el proceso de verificar los divisores retorna la cantidad acumulada.
- int divisor (int n, int dig)[**Operativa**]: Verifica la divisibilidad del número y el digito, retorna un dato booleano
- int ult_dig(int n)[**Operativa**]: Retorna el ultimo digito del valor numero, el numero n si tiene varios dígitos se va decrementando de forma imperactiva dividiendo por 10 el numero **n.**

[1] Universidad Tecnológica de Pereira

III.MACROALGORITMO

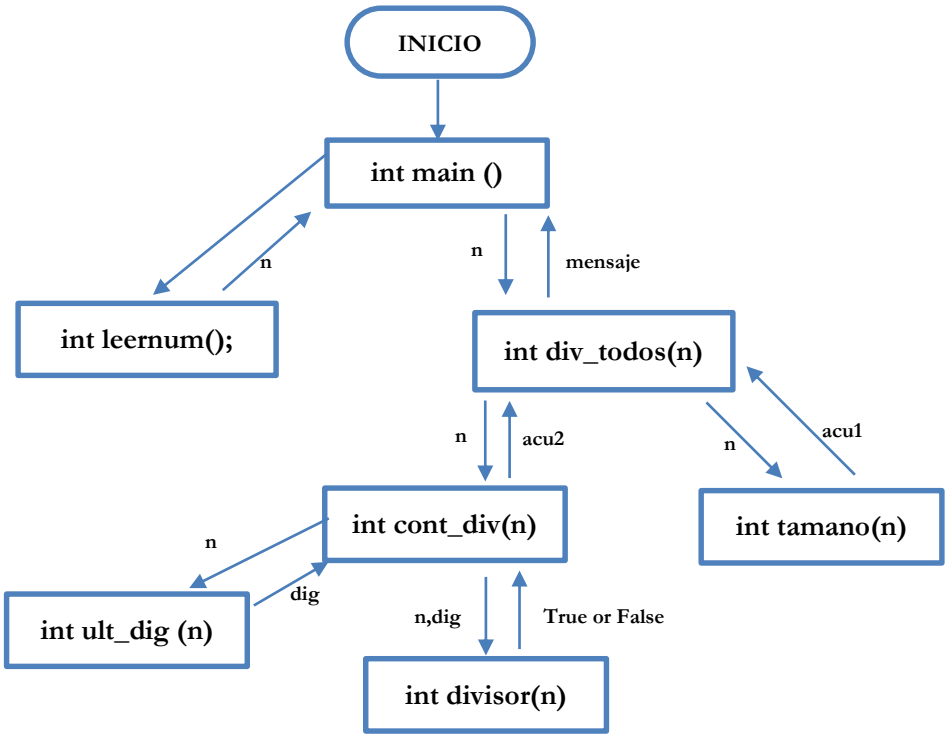
1. Leer un número entero ingresado por entrada estándar.
2. Determinar el tamaño del dato n y hallar la cantidad de dígitos.
3. Hallar el último digito de n.
4. Dividir el ultimo digito (*obtenido en 3*) sobre n, si esto da 0, incrementar un contador de divisores en 1.
5. Volver al paso 3, y actualizar el último digito.
6. Enviar la cantidad contada de dígitos que divide exactamente al número n del usuario, y mostrar el resultado por pantalla.

IV. CODIGO

Se encuentra anexo al correo electrónico con nombre de archivo **ejercicio1.cpp**

2. Leer un número ***Z*** y determinar si todos sus dígitos lo dividen exactamente.

I. DIAGRAMA FUNCIONAL



II. OBJETIVO DE FUNCIONES

- int leernum()[**Interface**]: Es una interface para obtener el numero **Z** que ingresa el usuario.
- int div_todos(int n)[**Interface-Operativa**]: Es una interface de usuario, esta función llama a dos subfunciones **cont_div** y **tamaño** n, si la cantidad de dígitos divide exactamente a al tamaño n, entonces envía un mensaje con una afirmación de que todos sus dígitos lo dividen exactamente.
- int cont_div(int n)[**Operativa**]: Se encarga de contar los divisores desde n hasta 0, cuando termina el proceso de verificar los divisores retorna la cantidad acumulada
- int divisor(int n, int dig)[**Operativa]:** Esta función es de tipo booleano , retorna un 1, si el numero n(ingresado por usuario) es divisible por el digito, de lo contrario retorna un 0 a la función cont_div.
- int ult_dig(int n)[**Operativa**]: se encarga de tomar el ultimo digito del valor ingresado por el usuario y retornarlo a la funcion divisor como segundo parámetro.
- int tamano(int n)[**Operativa**]: halla la cantidad de dígitos que contiene el numero entero.

III.MACROALGORITMO

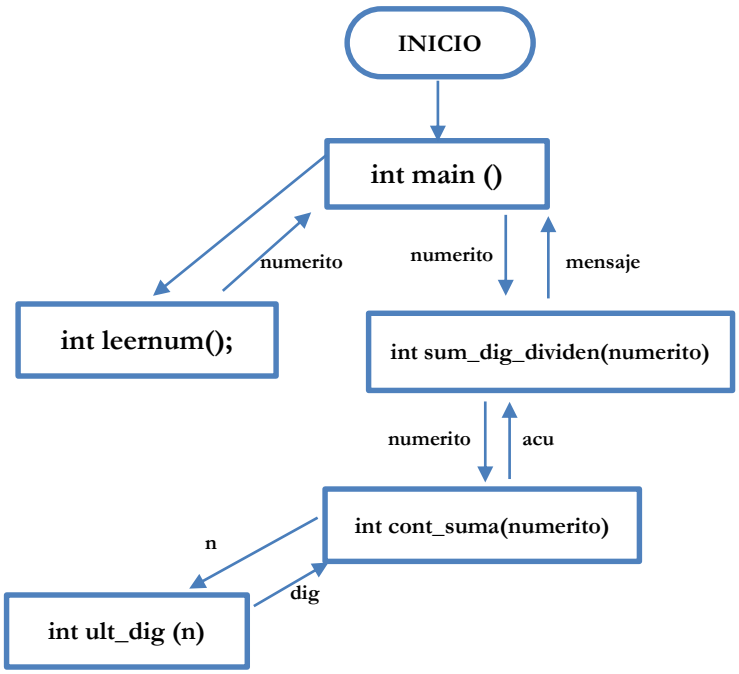
1. Leer un número entero ingresado por entrada estándar.
2. Que tamaño posee el dato n, cuantos dígitos.
3. Hallar el último digito de n.
4. Determinar que números son divisibles entre n, y último digito de n.
5. Contar la cantidad de dígitos que el número divide exactamente. Volver a **4** si el último digito no es 0.
6. Determinar si todos sus dígitos lo dividen exactamente, llamando a la función div_todos, comparando si todos los dígitos
7. Mostrar si todos sus dígitos lo dividen exactamente.

IV. CODIGO

El código se encuentra como anexo al correo electrónico, el archivo exacto es **ejercicio2.cpp**.

3. Leer un número ***Z*** y determinar si la suma de sus dígitos lo divide exactamente.

I. DIAGRAMA FUNCIONAL



II. OBJETIVO DE FUNCIONES

- int leernum()[**Interactiva**]: Es una interface para obtener el numero **Z** que ingresa el usuario.
- int sum_dig_dividen(int n)[**Operativa**]: Se encarga de verificar si la suma de los digitos del numero entero ingresado por el usuario dividen exactamente este numero.
- int cont_suma_dig(int n) [**Operativa**]: Realiza la suma de todos los digitos del numero entero ingresado por el usuario y lo retorna a las function sum_dig_dividen.
- int ult_dig(int n) **[Operativa]**: se encarga de tomar el numero n, e irlo descomponiendo en orden para obtener todos los dígitos del valor ingresado por el usuario, retornar el ultimo valor obtenido de hacer la operación n**mod**10 a la función divisor cont_suma.

III.MACROALGORITMO

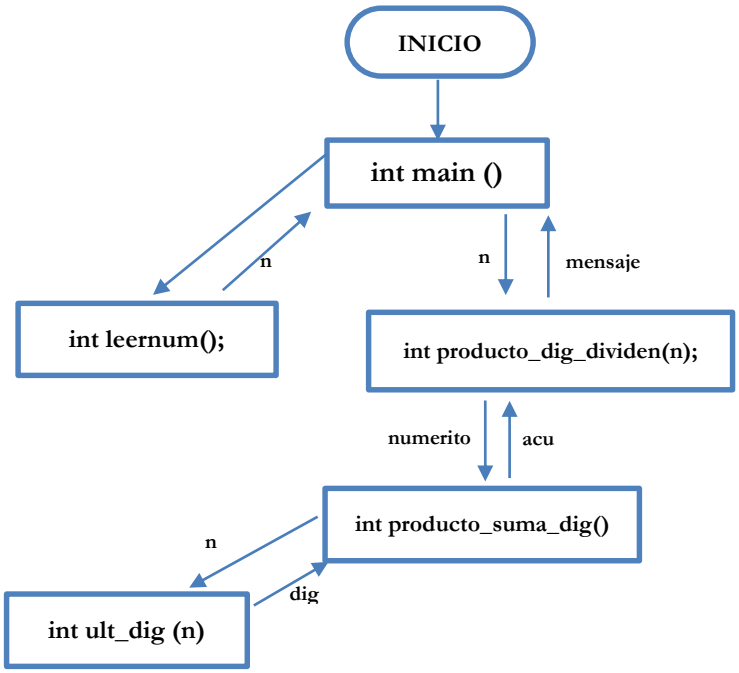
1. Leer un número entero ingresado por entrada estándar.
2. Obtener el ultimo digito de n
3. Contar la suma de los dígitos, sumado el digito obtenido en paso 2, hasta que el digito obtenido en paso dos sea cero.
4. Sumar los dígitos que dividen exactamente a n.
5. Mostrar por pantalla cuantos dígitos dividieron exactamente al número n ingresado por el usuario.

IV. CODIGO

El código se encuentra como anexo al correo electrónico, el archivo exacto es **ejercicio3.cpp**.

4. Leer un número ***Z*** y hallar si el producto de sus dígitos lo dividen exactamente

I. DIAGRAMA FUNCIONAL



II. OBJETIVO DE FUNCIONES

- int leernum()[**Interactiva**]: Es una interface para obtener el numero **Z** que ingresa el usuario
- int ult_dig(int n) **[Operativa]**: se encarga de tomar el numero n, e irlo descomponiendo en orden para obtener todos los dígitos del valor ingresado por el usuario, retornar el ultimo valor obtenido.
- int producto_suma_dig(int n) **[Operativa]**: Se encarga de realizar el producto de todos los dígitos del numero n.
- int producto_dig_dividen(int n)**[Operativa]**:Se encarga de realizar la división entre el producto obtenido ne la función producto_suma_dig y el numero ingresado por el usuario.

III.MACROALGORITMO

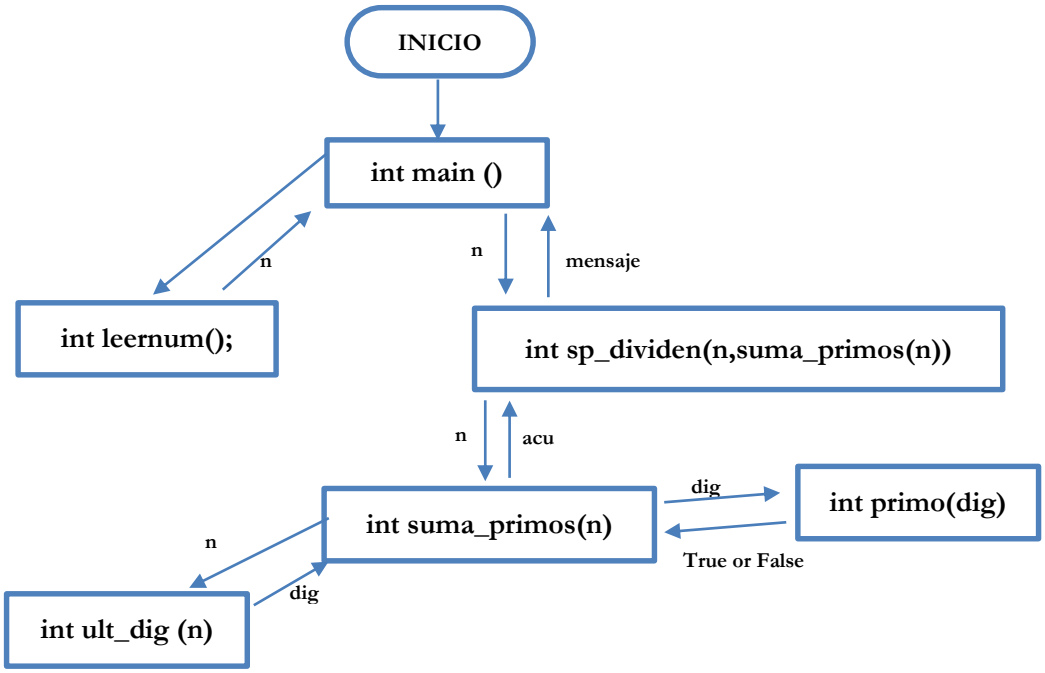
1. Leer un número entero ingresado por entrada estándar.
2. Obtener el último digito de n.
3. Realizar el producto de todos los dígitos de n.
4. Dividir el producto entre el digito y n, si es divisible mostrar en pantalla que el producto divide exactamente el digito n.

IV. CODIGO

El código se encuentra como anexo al correo electrónico, el archivo exacto es **ejercicio4.cpp**.

5. Leer un número ***Z*** y determinar si la suma de sus dígitos primos lo divide exactamente.

I. DIAGRAMA FUNCIONAL



II. OBJETIVO DE FUNCIONES

- int leernum()[**Interactiva**]: Es una interface para obtener el numero **Z** que ingresa el usuario
- int ult_dig(int n) **[Operativa]**: se encarga de tomar el numero n, e irlo descomponiendo en orden para obtener todos los dígitos del valor ingresado por el usuario, retornar el ultimo valor obtenido.
- int primo() **[Operativa]**: Determina si el digito es primo, de ser así retorna el digito, de lo contrario retorna 0.
- int suma_primos() **[Operativa]**: Esta función suma los dígitos que sean números primos.
- int sp_dividen(int n, int sum) **[Operativa]:** Esta funcion determina si la suma de los digitos primos divide exactamente al numero n

III.MACROALGORITMO

1. Leer un número n entero ingresado por entrada estándar.
2. Obtener el último digito de n.

3. Verificar si ese último digito es primo.

4. Tomar los números primos y sumarlos.

5. Realizar la división de la sumade primos con el numero n, si sumaprimos**mod**n es 0 enviar mensaje de respuesta afirmando esto.

IV. CODIGO

El código se encuentra como anexo al correo electrónico, el archivo exacto es **ejercicio5.cpp**.

CONCLUSIONES

Las funciónes nos permiten construir programas modulares, con los cuales podemos dividir un gran problema en subproblemas y la suma de las pequeñas soluciones resuelven la solución general. Además consiguen que no se repita el mismo código en varias partes del programa o software que estemos desarrollando: en lugar de escribir el mismo código cuando se necesite, por ejemplo para validar una fecha, se hace una llamada a la función que lo realiza.

C++ es un lenguaje extremadamente rico en funciones para el manejo de muchas estructuras de datos y tareas complejas, aprender a utilizar la rueda que ya esta usada es parte esencial del programador, además se debe documentar muy bien el código fuente para tener una referencia en miras a futuro.

Se reutilizo el programa que se desarrollo desde el primer enunciado y salio de una manera muy rápido, puede compilar el código bajo el compilador MingW.