

NAME  
Jedith Ciprian

PAGES  
3/6

SPEAKER/CLASS  
Programación, Mec

DATE - TIME  
11-05-2024

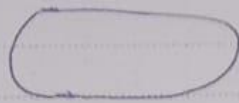
Title: Capítulo 3 Diagramas de flujo

Keyword

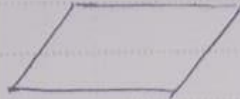
Estructura  
diagramas  
Optimización  
Planificación

Topic: Diagramas de flujo.

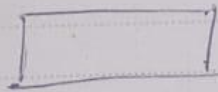
Notes: los diagramas de flujo son herramientas visuales que permiten planificar y estructurar el flujo de control de un programa antes de escribir el código en un lenguaje de programación como C.



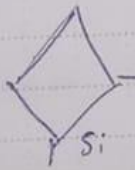
Inicio/Fin: Representado por un ovalo indica el inicio y el fin del diagrama de flujo.



Proceso: Representado por un rectángulo, indica una operación o un conjunto de operaciones que se lleva a cabo.



Entrada/Salida: Representado por un paralelogramo, indica la entrada de datos al programa o la salida de resultados.



no Representa un punto en el que se toma una decisión

Questions

Summary:

By Carlos Pichardo Vinque

NAME Judith Ciprian	PAGES 4/6	SPEAKER/CLASS Programación Mec	DATE - TIME 14-05-2024
------------------------	--------------	-----------------------------------	---------------------------

Title:

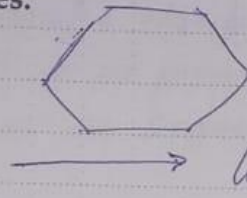
# Estructuras Repetitivas

Keyword

while  
do-while

Topic:

Notes:

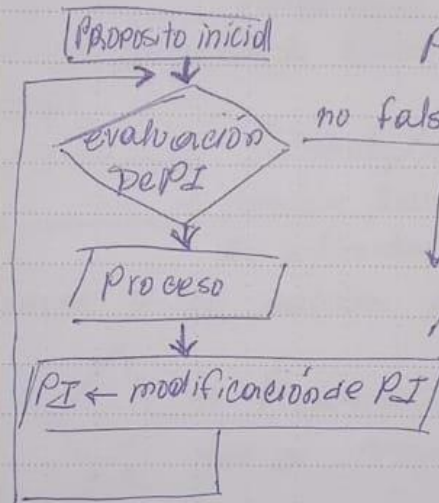


Representa una decisión

línea de flujo

La estructura "While" permite repetir un conjunto de instrucciones. El número de veces que se repite "while" depende de la condición que contenga el ciclo.

Questions



PI debe tener un valor diferente de cero y verdadero

dentro del ciclo debe existir un enunciado que afecte la condición

Summary:



NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Judith Ciprian	5/6	Programación Mec	14-05-2024

Title: Estructuras Algorítmicas

<b>Keyword</b> Algoritmo Diagrama de flujo ciclo Condición	<b>Topic:</b> En la resolución de problemas algorítmicos los ciclos se utilizan para repetir un conjunto de instrucciones un número determinado de veces. Existen tres estructuras algorítmicas representativas en el lenguaje C.  <b>Notes:</b> For se usa cuando se conoce de antemano el número de repeticiones. Las sintaxis incluyen una variable de control, un valor inicial, un valor final y un incremento o decremento.  <b>Questions</b>  <b>While:</b> Se usa cuando el número de repeticiones no se conoce previamente y depende de una condición que se evalúa al principio de cada iteración.  <b>Do-While:</b> Similar al While, pero la condición se evalúa al final, asegurando que el conjunto de instrucciones se ejecute al menos una vez.
--	---

Summary:

By Carlos Pichardo Vinque

NAME  
Judith Ciprian

PAGES  
6/6

SPEAKER/CLASS  
Programación Mec

DATE - TIME  
14-05-2024

Title:

Keyword

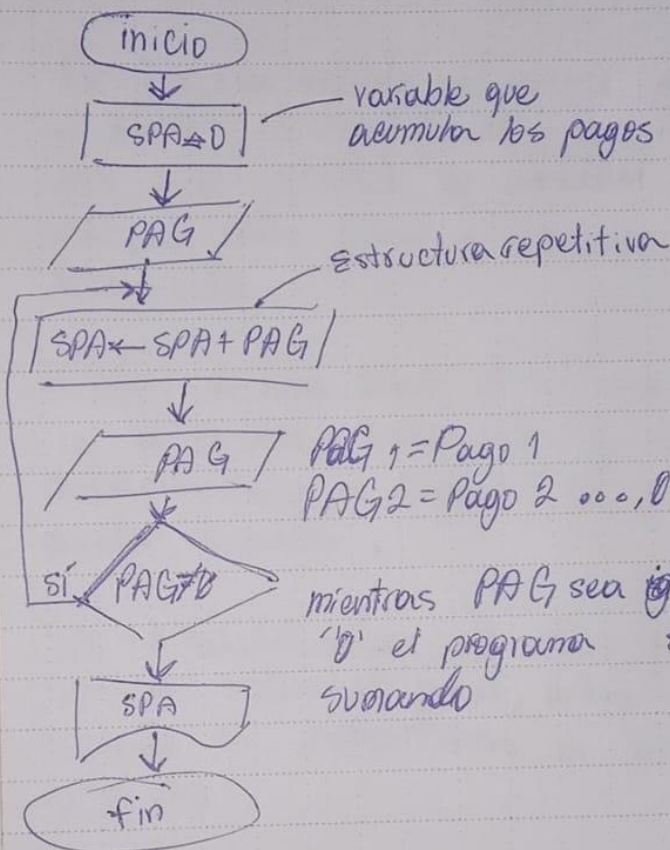
Estructuras  
Diagrama de  
Flujo

Topic: Estructuras Aritméticas

Notes: Uso de la estructura de-While  
Diagrama de flujo para la suma de pagos  
realizados el último mes.

Questions

¿Cómo ayudar  
en diagrama de  
flujo al proceso  
de programación?



Summary:

By Carlos Pichardo Viñue



NAME  
Judith Ciprian

CLASS  
1/4

SPEAKER  
Programacion Mec

DATE & TIME

31/5/2024

Title

Capítulo 4 Funciones

### Keyword

Funciones.  
Subproblemas.  
Reducción de  
problemas.  
Variables.  
Parámetros de  
valor.

### Topic

Para resolver problemas complejos en C, se descomponen en subproblemas mediante el uso de funciones, lo que facilita la lectura, escritura, mantenimiento del código y permite el trabajo en paralelo.

Un programa en C incluye un programa principal, generalmente breve que se llama a funciones y estas a su vez pueden llamar a otras funciones. La comunicación entre funciones y el programa principal se realiza mediante parámetros por valor, por referencia y menos comúnmente, variables globales. A diferencia de las funciones de bibliotecas estándar como `math.h`, las funciones desarrolladas en este contexto son creadas por el programador para resolver subproblemas específicos.

### Questions

¿Cuáles son las ventajas de utilizar funciones para resolver subproblemas?

**Summary:** la Mejor manera de resolver un problema en C, es descomponerlo en subproblemas utilizando funciones, lo que facilita la lectura, mantenimiento y trabajo en paralelo, y comunicación entre el programa principal y las funciones mediante parámetros por valor por referencia y mínimamente con variables globales.

NAME	CLASS	SPEAKER	DATE & TIME
Judith Ciprian	1/4	Programación Mec	31/05/2024

Title

Capítulo 4 Paso de Funciones como Parámetros

Keyword

Apuntadores  
Parámetros  
Cálculos

Topic

En la programación en C, es posible pasar funciones como parámetros por referencia, lo que permite una mayor flexibilidad en la resolución de problemas. Esta capacidad se logra utilizando apuntadores, que son variables que contienen direcciones de memoria. Por ejemplo en el programa presentado, se utilizan funciones como 'Suma' y 'Resta' las cuales son pasadas como parámetros a otra función 'control' que decide cual llamar según ciertas condiciones. Además, se resuelven varios problemas mediante uso de funciones

Questions

¿Que es un parámetro en C?  
Es una variable que se utiliza para enviar datos a una función cuando es llamada.

como determinar el número es múltiplo del otro, encontrar el mayor divisor común, contar números pares e impares, calcular las productoria de los primeros números naturales, entre otros. Estos problemas se resuelven mediante el paso de parámetros por valor y por referencia y se utilizan funciones auxiliares para realizar cálculos específicos.

**Summary:** la capacidad de pasar funciones como parámetros en C permite una programación más modular y eficiente para resolver una variedad de problemas.



## Capítulo 4 Variables locales, globales y estáticas.

**Inicialización** En C las variables se clasifican en locales, globales y estáticas. Las variables locales se definen dentro de funciones o bloques y solo existen mientras se ejecuta esa función o bloque, debiendo ser inicializadas cada vez que se usan. Tienen prioridad sobre las variables globales cuando comparten el mismo nombre.

**Tipo entero**  
**función**

**Variable estáticas**

**Parámetros**

Las variables globales, definidas antes del programa principal, son accesibles desde cualquier función. Las variables estáticas, aunque similares a las locales, conserva un valor entre llamadas a la función. Las variables estáticas, aunque similares

¿Qué es una variable global en "C"?

Variable que se declara fuera de cualquier función y puede ser accedida y modificada por todas las funciones en el programa

Las funciones en C tienen una estructura que incluye el tipo de resultado, el nombre de la función, los parámetros y el cuerpo de instrucciones, y pueden devolver valores específicos o no devolver ninguno, usando la palabra reservada "void" en este último caso.

NAME  
Judith Cipriani

CLASS  
4/4

SPEAKER  
Programación Mec

DATE & TIME  
31/05/2024

Title

## Capítulo 4 Variables y Parámetros.

### Keyword

Parámetros por valor

Apuntadores

Operadores de dirección

Comunicación entre funciones.

### Questions

¿Qué es una función en C?

Es un bloque de código que realiza una tarea específica y puede ser realizada en el programa.

**Topic** En C, la comunicación entre el programa principal y las funciones, o entre las funciones mismas se puede hacer mediante parámetros por valor o por referencia. Los parámetros por valor pasan una copia de la variable a la función, por lo que cualquier cambio en la función no afecta la variable original. Los parámetros por referencia, en cambio, pasan la dirección de la variable original usando apuntadores. Esto significa que los cambios en la función afectan directamente la variable original. Los apuntes se mantienen o manejan con los operadores de dirección (&) e indirección (\*).

Combinación de variables y parámetros

Un programa puede usar tanto variables locales como globales y parámetros por valor y por referencia.

Las variables locales tienen prioridad sobre las globales con el mismo nombre.

Las variables globales se pueden referenciar explícitamente usando "global".

**Summary:** el uso adecuado de estos conceptos permite un control preciso sobre el alcance y la modificación de variables en funciones; mejorando la claridad y eficiencia del código.