

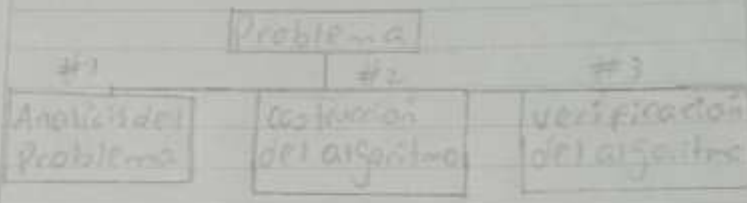
Title: Capitulo #1 Algoritmos, Diagramas de Flujo y Pseudocódigo

Keyword: Topic: Problemas y algoritmos.

Notes: Nosotros los seres humanos realizamos una serie de acciones para obtener o alcanzar, una o varias metas, estas acciones las realizamos hasta en nuestro día a día, vivir sin darnos cuenta de ello.

En esta figura podemos ver las etapas que podemos o debemos seguir para encontrar la solución.

Questions



Característica de dicha figura:

Precisión: se deben seguir de manera precisa
Determinismo: un algoritmo debe ser determinista
Finitud: un algoritmo debe ser finito.

Summary: En este capítulo habla de que nosotros los seres humanos, realizamos, varias acciones para obtener, metas y objetivos, realizadas o conseguidas con nosotros mismos, luego una figura que expresa un problema.


NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Daniel Pérez	82	P. 103	6/9/2024


Title: capítulo #1 Algoritmos, Diagramas de Flujo y Programación


Keyword: Topic: 1.2 Diagramas de Flujo

Notes: Un diagrama de Flujo es una representación gráfica de los pasos de un algoritmo para resolver un problema. Es esencial para diseñar el algoritmo antes de programar, y los símbolos utilizados siguen las normas.

Questions

 - se utiliza para marcar el inicio y el fin del diagrama de flujo.

 - se utiliza para introducir los datos de entrada. Expresa lectura.

 - Representa un proceso e incluye asignaciones, operaciones aritméticas y cambios de la memoria.

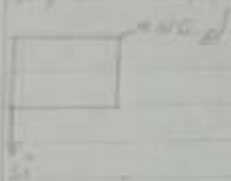
Summary: Como podemos ver la definición de un diagrama de flujo, que en pocas palabras, es un dibujo gráfico de los pasos de como se hace un algoritmo, y se presenta varias figuras importantes.

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Daylei Perez	#3	P.m	6/9/2024

Title: Capitulo #1 Algoritmos, Diagramas de Flujo y Pseudocódigo

Keyword: Topic 1.2 Diagramas de Flujo continuos

Notes: Representa una decisión basada en una condición, determinando qué camino seguir. Se usa en estructuras IF, IF-else, for, while y do-while.

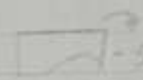


↓ ↑ → Expresa la dirección del flujo del diagrama.

Questions: Representa una decisión multiple en 'switch', con un selector que dirige el flujo según su valor.



O Expresa conexión dentro de una misma página



-se utiliza para representar la impresión de un resultado.

Summary: vemos una serie de dibujos que expresan sus definiciones, cada uno explicado para que sea, estas dibujos son muy interesantes, porque, nos explican funciones que desconocíamos que tenían.

NAME	PAGES	SPEAKER/CLASS	DATE-TIME
Darbel Pérez	#4	Kenn	6/19/2024

Title: Capítulo #1 Algoritmos, Diagramas de Flujo y

Keyword

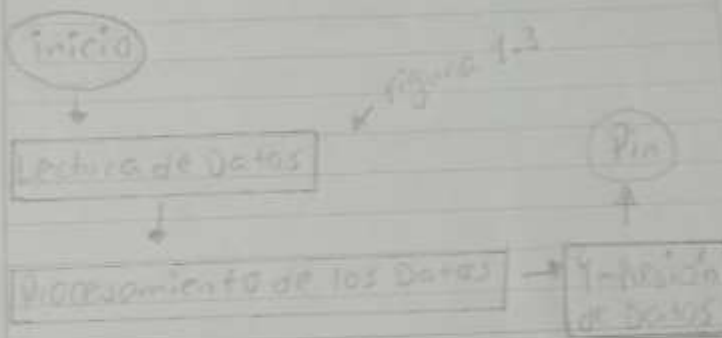
Topic: 1.2 Diagrama de Flujo. Continuación

Notes: Representa conexión entre páginas

→ Representa un subproblema que resolver antes de continuar con el flujo principal del diagrama.

Questions

La figura 1.3 muestra los pasos para construir un diagrama de flujo. El procesamiento de datos suele implicar toma de decisiones y es común repetir algunos casos.



Summary: En esta continuación vemos una breve definición de lo que es, / también, nos muestra, por figuras con sus respectivas explicaciones. En pocas palabras, es una explicación.

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
------	-------	---------------	-------------

DATE: 10/22/24
#5
P. 10
6/19/2024

Title: Capítulo #1 Algoritmos, Diagramas de flujo y Pseudocódigo

Keyword: Topic: Los Algoritmos para la toma de decisiones de flujo.
Notes: un diagrama de flujo es la forma para resolver un problema, traduciendo la estructura del problema en un lenguaje como los algoritmos para solucionar.

Questions: 1. Debe tener un inicio y fin.
2. Las líneas de flujo deben ser rectas conectadas.

3. Comienza de arriba hacia abajo y de izquierda a derecha.

4. La notación debe ser independiente del lenguaje.

5. Incluye comentarios para tareas complejas.

Summary: En este capítulo nos muestran los pasos con los pasos para hacer un diagrama de flujo. Entre estos pasos se encuentran: que debe tener un inicio y un fin, incluir comentarios para dichas tareas.

NAME Daniel Cruz	PAGES #6	SPEAKER/CLASS Prim	DATE-TIME 6/11/2024
---------------------	-------------	-----------------------	------------------------

Title: Cap, lab #1 Algoritmo, Diagrama de flujo y Programación

Keyword	Topic: 1.3 Tipos de datos												
Notes	<p>Los datos que procesa una computadora se dividen en dos categorías: simples y los datos estructurados.</p> <p>Datos simples: son aquellos que se almacenan solo una instancia de memoria. Estos son: Enteros, Reales, Carácter.</p> <p>Datos estructurados: se caracterizan por referirse a un conjunto de cosas de memoria con un solo nombre. Es decir que un dato estructurado tiene múltiples componentes como: arrays y registros.</p>												
Questions	<table border="1"> <thead> <tr> <th>tipo de dato</th><th>Descripción</th><th>Rango</th></tr> </thead> <tbody> <tr> <td>int</td><td>Enteros</td><td>32768 a 32767</td></tr> <tr> <td>float</td><td>Reales</td><td>2.4×10^{-45} a 3.4×10^{38}</td></tr> <tr> <td>long</td><td>Enteros de 64 bits</td><td>-2^{63} a $2^{63}-1$</td></tr> </tbody> </table>	tipo de dato	Descripción	Rango	int	Enteros	32768 a 32767	float	Reales	2.4×10^{-45} a 3.4×10^{38}	long	Enteros de 64 bits	-2^{63} a $2^{63}-1$
tipo de dato	Descripción	Rango											
int	Enteros	32768 a 32767											
float	Reales	2.4×10^{-45} a 3.4×10^{38}											
long	Enteros de 64 bits	-2^{63} a $2^{63}-1$											

Summary: En este capítulo, nos presentan y nos explican como se procesan los datos de una computadora, y que son los datos simples y como están caracterizados los datos.

NAME Daylen Perla	PAGES 47	SPEAKER/CLASS P.m	DATE - TIME 6/01/2024
----------------------	-------------	----------------------	--------------------------

Title: Capitulo #1 Algoritmos, Diagramas de Flujo y Pseudocódigo

Keyword	Topic: 1.1.1 Tipos de Datos.
Questions	Notes: Los datos en un computador se dividen en simples (como int, float, long, double, char) que ocupan una celda de memoria y estructuras (como arrays, cadena, registros) que abarcan multiples celdas.
	Los datos se almacenan en celdas de memoria con un identificador que empieza con una letra y puede incluir letras, dígitos y guiones bajos. C distingue entre mayúscula y minúscula y tiene palabras reservadas que no se pueden usar como identificadores.
	Las constantes son datos fijos que no cambian durante la ejecución del programa y se definen con "const" o "final" o "enum".

Summary: Este capítulo vemos como se dividen los datos de una computadora, como se caracterizan, y cuales son las constantes, y nos presenta un código y para que sirve como uso.

By Carlos Fernando Vargas

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
DARKLIPS	#7	P.m	6/19/2024

Title: Capítulo 1 Algoritmo, Diagrama de Flujos y Programación

Keyword	Topic: 1.3.1 Indicadores																																
Notes:	Los datos se almacenan en celdas de memoria con un nombre único (identificador) que comienza con una letra y puede incluir letras, dígitos y el carácter "_". En los identificadores distinguir entre mayúscula y minúscula (Ej., "Aux vs." Aux y surten diferente 3 y 7 caracteres. No se pueden usar palabras reservadas del lenguaje como identificadores.																																
Questions	<p>Tabla 1.3. Palabras reservadas de C++</p> <table><tr><td>auto</td><td>do</td><td>goto</td><td>long</td></tr><tr><td>break</td><td>default</td><td>if</td><td>sizeof</td></tr><tr><td>case</td><td>else</td><td>int</td><td>static</td></tr><tr><td>char</td><td>enum</td><td>long</td><td>struct</td></tr><tr><td>const</td><td>extern</td><td>register</td><td>switch</td></tr><tr><td>continue</td><td>float</td><td>return</td><td>typedef</td></tr><tr><td>default</td><td>for</td><td>short</td><td>union</td></tr><tr><td>unsigned</td><td>void</td><td>volatile</td><td>while</td></tr></table>	auto	do	goto	long	break	default	if	sizeof	case	else	int	static	char	enum	long	struct	const	extern	register	switch	continue	float	return	typedef	default	for	short	union	unsigned	void	volatile	while
auto	do	goto	long																														
break	default	if	sizeof																														
case	else	int	static																														
char	enum	long	struct																														
const	extern	register	switch																														
continue	float	return	typedef																														
default	for	short	union																														
unsigned	void	volatile	while																														

Summary: En este capítulo vemos que los datos se almacenan en una memoria que es la del PC, y que se guarda con un nombre único que lo hace distinto a los demás, vemos una tabla con las palabras reservadas.

By Carlos Federico Vargas

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Dorival Pérez	#9	P.m	6/9/2024

Title: Capítulo #1 Algoritmo, Diagrama de Flujo y Programación

Keyword	Topic: 1.3.2. Constantes
	Notes: Las constantes son datos cuyo valor no cambia durante la ejecución del programa. Puede ser cualquier tipo de dato (entero, real, carácter, cadena, etc.) y deben definirse antes de iniciar el programa principal.
	Usando "Const" se declara una constante con su tipo y valor:
Questions	<pre>Const int nu = 20; // Entero Const float rei = 2.18; // Real</pre>
	Usando #define, se define una constante si se especifica su tipo:
	<pre>#define nu 20 // Entero #define rei 2.18 // Real</pre>

Summary: En este capítulo verás varios códigos y como se deben usar, debemos saber que este código son del lenguaje de programación C, también veremos que son las constantes y su uso.

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Daybel Perez	#10	Y.m	6/9/2024

Title: Capitulo #1 Algoritmo, Diagramas de Flujos y Programación

Keyword	Topic: <u>1.3.3. variables</u>
Notes:	Las variables son elementos en un programa que pueden cambiar de valor durante la ejecución y se nombran usando identificadores. Pueden ser de cualquier tipo de dato, como enteros, reales o caracteres, y se declaran en el código principal o en funciones. Al asignar un nuevo valor a una variable, se reemplaza el valor anterior, lo que se conoce como una operación destructiva.
Questions	<p>Int u1 = 10; // Entero</p> <p>Float re1 = 3.25; // Real</p> <p>char ca1 = 't'; // Caracter</p> <p>También es posible asignar valores al momento de declarar la variable.</p> <p>Int u1 = 10, u2 = 15; Float re1 = 3.25, re2 = 6.485; char ca1 = 't', ca2 = 's';</p>

Summary: En este capítulo vemos que son las variables, que en pocas palabras son elementos o herramientas de un programa que puede cambiar su valor durante se este ejecutando dicho dato.

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Daniel Peña	#11	Paola	6/9/2024

Title: Capítulo #1 Algoritmo Diagrama de Flujo y Pseudocódigo

Keyword	Topic: 1.4 Operadores y 1.4.1 Operadores Aritméticos																		
	Notes: Los operadores permiten realizar diferentes tipos de operaciones en C. Existen operadores aritméticos, asignación, de incremento y decremento y el operador lógico.																		
Questions	<p>Operadores aritméticos permiten realizar cálculos entre operadores (números, constantes o variables) y el resultado siempre es un número.</p> <table border="1"> <thead> <tr> <th>Operación</th><th>Ejemplos</th><th>Resultados</th></tr> </thead> <tbody> <tr> <td>Suma</td><td>$x = 4.5 + 3;$</td><td>$x = 7.5$</td></tr> <tr> <td>Resta</td><td>$v = 4.5 - 1;$</td><td>$v = 3.5$</td></tr> <tr> <td>Multiplicación</td><td>$x = 4.5 * 3;$</td><td>$x = 13.5$</td></tr> <tr> <td>División</td><td>$x = 4 / 2;$ $x = 4.0 / 2.0;$</td><td>$x = 2$ $x = 2$</td></tr> <tr> <td>modulo (residuo)</td><td>$x = 15 \% 2;$ $v = (25 \% 2) / 2;$</td><td>$x = 1$ $v = 0.0$</td></tr> </tbody> </table>	Operación	Ejemplos	Resultados	Suma	$x = 4.5 + 3;$	$x = 7.5$	Resta	$v = 4.5 - 1;$	$v = 3.5$	Multiplicación	$x = 4.5 * 3;$	$x = 13.5$	División	$x = 4 / 2;$ $x = 4.0 / 2.0;$	$x = 2$ $x = 2$	modulo (residuo)	$x = 15 \% 2;$ $v = (25 \% 2) / 2;$	$x = 1$ $v = 0.0$
Operación	Ejemplos	Resultados																	
Suma	$x = 4.5 + 3;$	$x = 7.5$																	
Resta	$v = 4.5 - 1;$	$v = 3.5$																	
Multiplicación	$x = 4.5 * 3;$	$x = 13.5$																	
División	$x = 4 / 2;$ $x = 4.0 / 2.0;$	$x = 2$ $x = 2$																	
modulo (residuo)	$x = 15 \% 2;$ $v = (25 \% 2) / 2;$	$x = 1$ $v = 0.0$																	

Summary: En este capítulo vemos los operadores aritméticos es decir: la suma, la resta, multiplicación, división, también lo que nos permite hacer dichos operadores en algún programa.

NAME	PAGES	SPEAKER/CLASS	DATE-TIME
Daybel Pérez	#12	P.m	6/9/2024

Title: Capítulo #1 Algoritmos, Diagrama de Flujo y Programación "C"

Keyword	Topic: 1.4.2. operadores aritméticos simplificados
Notes:	En "C" los operadores aritméticos pueden usarse en formas simplificadas. Para hacer el código más conciso. Por ejemplo, en lugar de $x = x + y$, se puede usar $x += y$. Estas formas abreviadas facilitan la escritura y lectura del código.
Questions	Operadores aritméticos: Forma simplificada de uso. Ejemplos. Equivalencia Resultado $+ / + = x = 6;$ $x = 6$ $x = 6$ $+ / + = y = 4;$ $y = 4$ $y = 4$ $+ / + = x + 5;$ $x = x + 5;$ $x = 11$ $x / + = x + y;$ $x = x + y$ $x = 5$ $- / - = 10;$ $x = 10$ $y = 10$ $- / - = 5;$ $y = 5$ $y = 5$ $- / - = x - 3;$ $x = x - 3;$ $x = 7$ $- / - = y - x;$ $x = x - y;$ $x = 2$

Summary: En este capítulo vemos varios operadores aritméticos de una forma simplificada y nos muestra sus diferentes usos, también, nos explican los diferentes usos que puede tener.

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Daynel Perez	#17	V. n7	6/9/2024

Title: Capítulo #1 Algoritmo, Diagrama de flujo y Programación en C

Keyword	Topic: 1.4.3 Operadores de Incremento y Decremento
	Notes: Los operadores de incremento (++) y decremento (--), en C son útiles porque simplifican y aclaran la estructura de programas. Pueden aplicarse antes o después de una variable, lo que produce diferentes resultados.
	Operador Operación Ejemplo Resultado
	++ Incremento $x=7;$ $x=7$
	$y=x++;$ $y=7$
	$x=8$
Questions	$x=7;$ $x=7$
	$y=++x;$ $y=8$
	$x=8$
	-- Decremento $x=6$ $x=6$
	$y=x--;$ $y=6$
	$x=5$
	$x=6;$ $x=6$
	$y=--x;$ $y=5$
	$x=5$

Summary: En este capítulo veremos, más operadores, y veremos sus diferentes resultados, también veremos que son los operadores de incremento, y para que se utilizan y veremos otros ejemplos.

NAME	PAGES	SPEAKER/CLASS	DATE-TIME
Daniel Perez	24	Plan	6/9/2024

Title: Capítulo 1 Algoritmo, Diagrama de flujo y programación

Keyword	<p>Topic: 1.4.4. Expresiones lógicas y L.L.S</p> <p>Operadores relacionales</p> <p>Notes: Las expresiones lógicas o booleanas en C están formadas por un número, constantes, variables y operadores lógicos o relacionales, y pueden ser verdaderas (1) o falsas (0).</p> <p>Los operadores relacionales se emplean para comparar dos operandos (números, caracteres, cadenas, constantes o variables).</p>
Questions	<p>Operador Operación Ejemplos</p> <p>relacional == igual si $a=1$; $res=0$</p> <p>!= diferente si $a=1$; $res=1$</p> <p>< menor que si $a=7$; $b=15$; $res=1$</p> <p>> mayor que si $a=22$; $b=11$; $res=1$</p> <p><= menor o igual que si $a=15$; $b=20$; $res=0$</p> <p>>= mayor o igual que si $a=20$; $b=22$; $res=1$</p> <p>res = $(7 > 15) ? 0 : 1$; $(22 > 11) ? 1 : 0$</p>

Summary: En este capítulo vemos que son las expresiones lógicas, y porque están formadas, vemos más operadores, pero aquí explican cada función, y vemos que están formadas por números.

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
DANIEL PEREZ	#15	P. m	6/9/2024

Title: Capítulo 2. Algoritmos, Diagrama de flujo y Programación

Keyword: Topic: 1.4.6. Operadores lógicos

Notes: Los operadores lógicos en C permiten crear condiciones complejas a partir de condiciones simples.

Operador lógico / operación / expresión / Resultado

!	negación	$x \neq 1$	$x = 1$
&&	conjunción	$x \neq 0$ y $y = 1$	$x = 1$
		$x \neq 0$ y $y = 0$	$x = 0$
		$x \neq 0$ y $y = 1$	$x = 1$
		$x \neq 0$ y $y = 0$	$x = 0$

Questions

Tabla de verdad de los operadores lógicos:

P	Q	(!P)
Verdadero	Verdadero	Falso
1	1	0
Verdadero	Falso	Falso
1	0	0
Falso	Verdadero	Verdadero
0	1	1
Falso	Falso	Verdadero
0	0	1

Summary: En este capítulo podemos observar que en C permiten crear unas condiciones complejas, también vemos una tabla con varios operadores lógicos y algunos ejemplos mostrados.

NAME Diana Perez	PAGES #17	SPEAKER/CLASS Car	DATE-TIME 6/9/2024
---------------------	--------------	----------------------	-----------------------

Title: Capítulo 1.1. Aritmética, Diagrama de flujo y po-
sición

Keyword	Topic: 1.1.1. Propiedades de las opera- ciones
Notes	En el capítulo 1.1 se discuten las propiedades de las operaciones aritméticas. En este capítulo se discuten las propiedades de las operaciones aritméticas. En este capítulo se discuten las propiedades de las operaciones aritméticas.
Questions	<p>Operadores: $+$, $-$, \times, \div, $\%$</p> <p>Precedencia: $+$, $-$, \times, \div, $\%$</p> <p>Asociatividad: $(a+b)+c = a+(b+c)$</p> <p>Commutatividad: $a+b = b+a$</p> <p>Asociatividad: $(a \times b) \times c = a \times (b \times c)$</p> <p>Commutatividad: $a \times b = b \times a$</p> <p>Asociatividad: $(a \div b) \div c = a \div (b \div c)$</p> <p>Commutatividad: $a \div b = b \div a$</p> <p>Asociatividad: $(a \% b) \% c = a \% (b \% c)$</p> <p>Commutatividad: $a \% b = b \% a$</p>

Summary: En este capítulo se discuten las propiedades de las operaciones aritméticas. En este capítulo se discuten las propiedades de las operaciones aritméticas. En este capítulo se discuten las propiedades de las operaciones aritméticas.

NAME	PAGES	SPEAKER/CLASS	DATE-TIME
------	-------	---------------	-----------

Dante Pérez

#11

Pam

6/12/2024

Title: Capítulo 1 Algoritmo Diagrama de flujo

Keyword	Topic: 1.5. Construcción de diagrama de flujo
Notes	<p>Un diagrama de flujo es una representación gráfica de un algoritmo y se utiliza para escribir programas en lenguaje de programación.</p> <p>Construye un diagrama de flujo que, al recibir los datos A, B, C y D que representen números enteros, elimine los números en orden inverso.</p>
Questions	<p>Inicio</p> <p>↓</p> <p>A, B, C, D ¿Se leen los datos?</p> <p>↓</p> <p>D, C, B, A ¿Se escriben los datos en orden inverso?</p> <p>↓</p> <p>Fin</p>

Summary: En este capítulo posemos un diagrama de flujo, y muestra como se construye dicho diagrama, se des explica las funciones que tienen los de los pasos del diagrama de flujo.

Ey (Bilder) Federico Piquero

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Daybel Pérez	#14	P.m	6/9/2024

Title: Capítulo #1 Algoritmos, Diagramas de Flujo y Programación

Keyword	Topic: Diagrama de flujo p+2 cont. noción
	Notes: Diagrama de flujo: recibe la clave de un empleado y sus seis primeros sueldos de año. Calcula el ingreso total y el promedio mensual, e imprime estos valores.
Questions	<p>(Inicio)</p> <p>↓</p> <div style="display: flex; justify-content: space-between;"> <div style="border: 1px solid black; padding: 5px;"> <p>CLA, SU1, SU2 SU3, SU4, SU5, SU6</p> <p>↓</p> <p>$Inc = (SU1 + SU2 + SU3 + SU4 + SU5 + SU6)$ $Pro \leftarrow Inc / 6$</p> </div> <div> <p>/* Lectura de la clave del empleado y los sueldos percibidos */</p> <p>/* Cálculo de ingreso total y del promedio */</p> </div> </div> <p>CLA, Inc, Pro</p> <p>↓</p> <p>(Fin)</p> <p>/* Escribe la clave del empleado, el ingreso total y el promedio */</p>

Summary: En este capítulo podemos observar un diagrama de flujo, también podemos ver algunos códigos que se pueden aplicar para poder hacer algunos cálculos muy interesantes.

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Davinci Pérez	#20	P.m	6/9/2024

Title: Capítulo #1 Algoritmo, Diagrama de flujo y programa "C"

Keyword: Topic: 1.6. Programas

Notes: Un programa fue un concepto desarrollado por van Neuman en 1946, es un conjunto de instrucciones que una computadora sigue para lograr un resultado específico. Se escribe en un lenguaje de programación, como C, a partir de un diagrama de flujo diseñado previamente.

Questions: El lenguaje de programación está compuesto por reglas sintácticas (formación de instrucciones) y semánticas (significado de las instrucciones).
"C" es un lenguaje estructurado que implementa de manera organizada y facilita la comprensión, depuración y modificación del programa.

Summary: En este capítulo vemos la definición de un programa que fue creado por una persona llamada van Neuman. Se describe en un lenguaje de programación, como es "C".

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Daybel Perez	#21	P.m	6/19/2024

Title: Capítulo #1 Algoritmo, Diagrama de Flujo y Programa en

Keyword	Topic: 1.6.1 Caracteres de control
Questions	Notes: Los caracteres de control son símbolos utilizados para dar forma y ajustar como presentan los resultados impresos en pantalla. Cada carácter de control tiene un efecto específico, como saltos de líneas, tabulaciones.
	Tabla 1.15 caracteres de control
	Carácter de control Explicación
	<div> <div>ln</div> <div>Permite usar a una nueva línea.</div> </div> <div> <div>lt</div> <div>Permite tabular horizontalmente.</div> </div> <div> <div>lv</div> <div>Permite tabular verticalmente.</div> </div>

Summary: En este capítulo vemos que son los caracteres de control, y una tabla que explica las funciones de cada símbolo, y vemos que cada carácter tiene su propio efecto en el programa.

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
------	-------	---------------	-------------

Daniel Pineda

#22

P. 1

6/9/2024

Title: Capitulo #1 de C, Diagrama de flujo y Pseudocódigo

Keyword

Topic: 1.6.2 Formato de salida en C

Notes: En el lenguaje de programación C, el formato de una lectura y escritura de variables varían según su tipo de datos, específicamente el formato correcto es esencial al usar las funciones (scanf y printf).

Questions

Por ejemplo se utiliza para enteros %d, %i, %o, %u, %x, %X para enteros, %f para números reales, %e para caracteres, usando estas formattas garantizo que los datos se manejan y se representen correctamente durante la ejecución del programa.

Ejemplo de variables:
float x = 6.355, z = 7.2576
int y = 4, t = -5;

Summary: En C es crucial usar formato para leer y escribir variables (con scanf y printf), ya que depende el tipo de dato. Por ejemplo %d es para enteros sin signo, %i para enteros, %f para números reales.

NAME	PAGES	SPEAKER/CLASS	DATE-TIME
Darlene Perez	#1	P.m	6/9/2024

Title: Capítulo #1 Sistema Numérico

Keyword: Topic: 1.1 Introducción

Notes: A lo largo de la historia, se han desarrollado diferentes sistemas de numeración para representar cantidades.

Questions

Los primeros sistemas, como el egipcio y el romano, eran aditivos, sumando valores de símbolos sin importar su posición. Sin embargo, estos sistemas eran poco prácticos.

Los babilonios introdujeron uno de los primeros sistemas posicionales, basado en el número 60, que aún se usa para medir tiempo. Aunque tenía problemas con el cero, los mayas, en cambio, crearon un sistema posicional de base 20.

Summary: Históricamente, se usaron sistemas aditivos y posicionales. Los mayas introdujeron el concepto de cero. Hoy se utilizan sistemas posicionales como decimal, binario, octal y hexadecimal.

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Daniel Peres	#2	P. 00	6/9/2024

Title: Capítulo #1 Sistemas numéricos

Keyword	Topic: 1.2 Sistemas decimal
	Notes: Sistema decimal se basa en diez dígitos (0-9) y emplea la notación posicional para representar valores. Cada dígito en un número tiene un valor que depende de su posición y de potencias de 10. En el número 83674, los dígitos tienen los siguientes valores potenciales:
Questions	<p>parte entera: $8(10^4)$, $3(10^3)$, $6(10^2)$</p> <p>parte fraccionaria: $7(10^{-1})$, $4(10^{-2})$</p> <p>La forma exponencial para este número es $8 \times 10^4 + 3 \times 10^3 + 6 \times 10^2 + 7 \times 10^{-1} + 4 \times 10^{-2}$. Este método facilita la conversión entre diferentes sistemas numéricos y asegura una representación.</p>

Summary: El sistema decimal usa diez dígitos (0-9) y notación posicional para representar valores. En el número 83674, cada dígito tiene su valor basado en su posición y potencias de 10.

By Carlos Federico Viquez

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
David Jones	#4	P.m.	6/17/2024

Title: Capítulo #1 Sistema Numéricos

Keyword	Topic: 1.3.3 sistema hexadecimal
Notes	<p>El sistema usa base de 16 y emplea los dígitos 0-9, junto con los 10 A-F, números enteros de 0 a 15.</p> <p>Ejemplo:</p> <p>Para convertir el número hexadecimal E8A7.3D a decimal, primero se convierte a decimal.</p>
Questions	<p>$E8A7.3D_{16} = 56556.2321_{10}$</p> <p>Luego, se convierte el decimal a binario, obteniendo una representación binaria aproximada debido a errores de redondeo.</p> <p>En la comparación entre sistemas, hexadecimal con un bloque de 4 bits en binario.</p>

Summary: El sistema hexadecimal usa una base de 16 con dígitos de 0-9 y letras A-F (10 a 15). Cada dígito tiene un valor basado en potencias de 16 para convertir a decimal.

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Daniel Pérez	# 5	Perez	6/9/2024

Title: Capítulo #1 Sistema numérico

Keyword: Topic: 1.4 Generalización de los sistemas numéricos

Notes: En la misma forma en que creamos los sistemas posicionales como el decimal, binario, octal y hexadecimal, se pueden desarrollar sistemas posicionales personalizados usando dígitos y letras según sea necesario. Por ej:

Questions

Base 7: El sistema usa dígitos del 0 al 6, como en el número 20541321.

Base 16: permite dígitos del 0 al 9 y letras A a F (10 a 15), como en el número 305A92FD148.

Cada sistema posicional tiene un valor máximo que es la base menos uno para sus dígitos.

Summary: Los sistemas posicionales personalizados pueden usar dígitos y letras según la base deseada.

Base 7: usa dígitos del 0 al 6 (e.g. 20541321).

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Daniel Perez	#1	1m	6/9/2024

Title: Capítulo # 2 Métodos de Conteo

Keyword

Topic: 2.1 Introducción

Notes: Contar es esencial en diversos contextos, desde evaluar el clima y estadísticas demográficas hasta calcular la población de producción y controlar inventarios.

Questions

En computación, los métodos de conteo ayudan a medir la eficiencia de programas, como el número de ciclos, comparaciones o intercambios que realizan.

un software puede considerarse eficiente si se ejecuta menos complicaciones o ciclos para complementar una tarea, de comparado con otros. Así, el uso adecuado de estos optimiza.

Summary: Es posible contar diversos elementos, como el dinero, la población de un grupo específico, o el número de combinaciones posibles para placas vehiculares, usando métodos de conteo.

NAME	PAGES	SPEAKER/CLASS	DATE-TIME
Daniel Pérez	#2	P. m	6/9/2024

Title: Capítulo #2 métodos de conteo

Keyword	<p>Topic: 2.2 Principios fundamentales del conteo</p> <p>Notes: Los métodos de conteo en matemáticas utilizan los principios fundamentales: el principio fundamental del producto y el principio fundamental de la adición.</p>
Questions	<p>Principio fundamental del producto: Si una operación se puede realizar de n formas y cada una de estas puede llevarse a cabo de m maneras diferentes en una segunda operación, entonces la operación, entonces las operaciones.</p> <p>Principio fundamental de la adición: Si un evento puede ocurrir en n maneras distintas o en m lugares distintos, y no puede ocurrir en dos lugares.</p>
Summary:	<p>Los métodos de conteo se basan en: principio del producto: si hay forma de hacer algo y cada una se puede hacer de m maneras, el total es $n \times m$. Principio de Adición: si un evento.</p>

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
José Luis P. C.	#3	1er	07/02/2024

Title: Capítulo #3 métodos de conteo

Keyword

Topic: 2-3 permutaciones

Notes: Las permutaciones son formas de ordenar objetos donde es importante la combinación. Estudia ordenaciones de objetos según ciertos criterios, y se aplica en diversos campos como algebra, probabilidad y computación.

Questions

Permutaciones: número de arreglos posibles de un conjunto donde el orden importa. Se calcula usando el factorial del número de elementos (n!). Ejemplos:

Para 3 maestros en 3 puestos diferentes, hay $3! = 6$ permutaciones.

Combinatoria: Analiza la cantidad de manera de seleccionar y ordenar.

Summary:

Las permutaciones son diferentes formas de ordenar un conjunto de objetos donde el orden importa. Calcula mediante el factorial del número de elementos (n!). Por ejemplo, con 3 maestros y 3 puestos.

Title: Capitulo #2 método de conteo

Keyword

Topic: 2.4 Combinaciones

Notes: Las combinaciones se refieren a la selección de elementos de un conjunto sin importar el orden en que se eligen. Se calcula usando fórmula.

$${}^n C_r = \frac{n!}{r!(n-r)!}$$

Questions

Donde n es el total de elementos y r es el número de elementos seleccionados.

Ej. seleccionar 2 maestros de un grupo de 3 (sin importar el puesto).

Hay solo 1 combinación ya que todos los elementos se usan y el orden no importa.

Summary: La combinación se refiere a la selección de un subconjunto de elementos de un conjunto mayor, sin que el orden de selección importe, para calcular el número de combinaciones posibles.

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
------	-------	---------------	-------------

Daniel Perez

#5

P.m

6/9/2024

Title: Capítulo #2 métodos de conteo

Keyword	Topic: 2.5 Aplicaciones en la combinatoria
	Notes: En combinatoria y matemáticas, los métodos de conteo se aplican para resolver problemas relacionados con la cantidad de ejecuciones de instrucciones, la representación de datos, y otros temas similares.
Questions	<p>En caso notarse que el binomio elevado a un binomio $(x+y)^n$ la potencia n, se utiliza la fórmula del binomio de Newton para expandirlo, la fórmula general para $n \geq 0$ es:</p> $(x+y)^n = \sum_{k=0}^n \binom{n}{k} x^{n-k} y^k$ <p>Donde, $\binom{n}{k}$ son los coeficientes binomiales, tal como se ve:</p>

Summary: En combinatoria y matemáticas, los métodos de conteo ayudan a resolver problemas como contar ejecuciones de instrucciones y representar datos.

$$(x+y)^n = \sum_{k=0}^n \binom{n}{k} x^{n-k} y^k$$

