ESTRUCTURAS DE PROGRAMACIÓN

CICLOS

**Ciclo para (For)**

Se llaman problemas repetitivos o cíclicos a aquellos en cuya solución es necesario utilizar un mismo conjunto de acciones que se puedan ejecutar una cantidad específica de veces. Esta cantidad puede ser fija (previamente determinada por el programador) o puede ser variable (estar en función de algún dato dentro del programa).Los ciclos se clasifican en:

* Ciclos con un Número Determinado de Iteraciones [Para – Hasta] (iteraciones es la cantidad de veces que se ejecuta un ciclo).

Son aquellos en que el número de iteraciones se conoce antes de ejecutarse el ciclo. La forma de esta estructura es la siguiente:

|  |  |
| --- | --- |
| Para i=1 hasta i=10 [límite]  Acción 1  Acción 2  Etc..  i = i+1  Fin para | La variable de conteo necesita un valor inicial y uno final.  La cantidad de acciones o comandos dentro del ciclo pueden ser muchas  Se incrementa la variable de conteo  Finaliza la estructura. |

Donde:

i Variable de control o de conteo del ciclo

En este ciclo la variable de conteo toma el valor inicial del ciclo y el ciclo se repite hasta que la variable de conteo llegue a su límite.

Para entender el concepto de variables de conteo, se necesita conocer el concepto de variables contadores.

* Contadores: Se utilizan para llevar el control del número de ocasiones en que se realiza una operación o se cumple una condición. Con los incrementos generalmente de uno en uno, deben tener un valor inicial de 0 o 1 por lo general.

Ej: i=0 valor inicial 0

i=i+1 a *i* se le estará incrementando en 1 su valor.

**Ejercicios:**

1. Construya un algoritmo que reciba como entrada un número N, y de cómo salida N veces la palabra “hola”.

Inicio

Ingrese número N N en este caso será el límite

Para i=1 hasta N

Imprimir (hola)

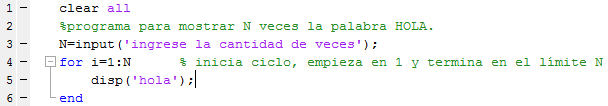
i=i+1

Fin para

Fin

Realizar la prueba de escritorio para comprobar los resultados.

Código en Matlab



**Nota:** el ciclo FOR en MATLAB no necesita incrementar la variable de conteo, ya que el comando lo realiza automáticamente.

1. Mostrar en pantalla los números del 1 al 10.
2. Dado un límite N, mostrar los números de forma consecutiva hasta dicho límite.

P.E N=6 en pantalla deberá salir 1 2 3 4 5 6.

1. En una planta de producción, existe un proceso que necesita tener su temperatura monitoreada de forma permanente, se debe ingresar una cantidad límite de valores de temperatura, y luego se deben ingresar los valores de las temperaturas que ha obtenido el sensor, el sistema debe mostrar una alerta en pantalla cada vez que la temperatura sea cero (0).

P.E:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | °C |  |
| Cantidad Límite de temperaturas | 5 |  | Valores de temperatura | 1 | -3 |  |
|  |  |  |  | 2 | 0 | ¡Temperatura Cero! |
|  |  |  |  | 3 | -1 |  |
|  |  |  |  | 4 | 0 | ¡Temperatura Cero! |
|  |  |  |  | 5 | 2 |  |

Se debe construir un programa que permita ingresar cualquier límite de y valores de temperaturas y verificar cuándo estas sean cero (0).

1. Leer 10 números y obtener el cubo y el cuadrado de cada uno de ellos a la vez.
2. Mostrar solamente los positivos de 5 números ingresados.
3. Ingresar 12 números, los negativos convertirlos a positivos y mostrar dichos números en pantalla.
4. Suponga que se tiene un conjunto de calificaciones de un grupo de 12 estudiantes. Realizar un algoritmo para calcular la calificación promedio de cada uno de ellos, solo dos notas por estudiante.
5. Modificar el ejercicio para que sirva para N estudiantes.
6. Mostrar el cubo de los números hasta un límite ingresado:

|  |  |  |
| --- | --- | --- |
| Lim= | 4 |  |
|  |  |  |
| 1 | ^3 | 1 |
| 2 | ^3 | 8 |
| 3 | ^3 | 27 |
| 4 | ^3 | 64 |

1. Escribir un programa para que muestre el siguiente patrón:

1

12

123

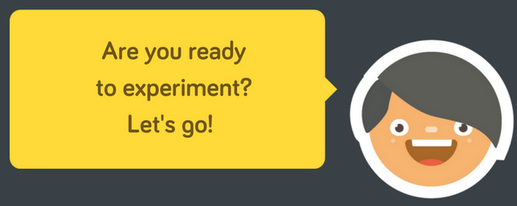
1234

El programa tiene como entrada la cantidad de filas que se desean, y por cada fila debe aumentar un número.

1. Dado un número como entrada, mostrar por cuales es divisible.
2. Ingresar por teclado N números no consecutivos y decir cuales son pares.
3. Hacer un algoritmo que le permita al usuario escoger un número N e imprimir su tabla de multiplicar.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | 5 |  |  |  |
|  |  |  |  |  |
|  | 5 | \* | 1 | 5 |
|  | 5 | \* | 2 | 10 |
|  | 5 | \* | 3 | 15 |
|  | 5 | \* | 4 | 20 |
|  | 5 | \* | 5 | 25 |
|  | 5 | \* | 6 | 30 |
|  | 5 | \* | 7 | 35 |
|  | 5 | \* | 8 | 40 |
|  | 5 | \* | 9 | 45 |
|  | 5 | \* | 10 | 50 |

**CONTADORES Y ACUMULADORES**



|  |  |
| --- | --- |
| **Contadores**: es una variable en la memoria que se incrementará en una unidad o una constante cada vez que se ejecute el proceso. |  |

Contador = contador+1

Contador = contador+2

Contador = contador+5

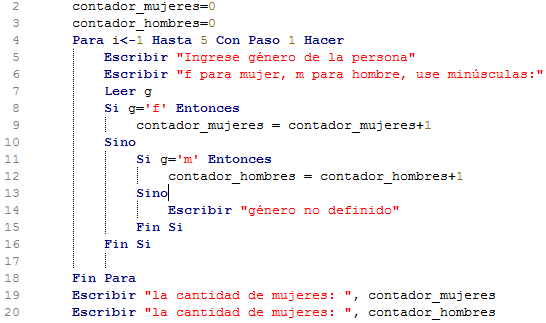
El contador se utiliza para llevar la cuenta de determinadas acciones que pueden ocurrir durante la solución de un problema o ejecución de una tarea, por ejemplo: contar la cantidad de personas que ingresan a una fiesta, o contar la cantidad de veces que una condición es cierta.

Los contadores, por lo general, se inicializan por fuera del ciclo, y consiste en poner el valor inicial de la variable que representa al contador en 0.

Ejemplo elaborado en Pseint:

Programa para contar la cantidad de mujeres y de hombres que se registran ingresando el género.

Nota: lo siguiente es una imagen, es inútil copiar y pegar el texto, he he he



En las líneas 2 y 3 se inicializan los contadores en cero, se debe crear un contador para cada cosa que se desee contar.

Línea 4, se inicia el ciclo desde 1 hasta 10, con i como variable de conteo.

En la línea 7 se ingresa en género de la persona, va dentro del ciclo porque se pregunta 10 veces.

Línea 8, se pregunta con un condicional si el género es ‘f’, si es así, en contador de mujeres se incrementará en 1, de igual forma para los hombres.

Las líneas 19 y 20 son para mostrar los resultados de los contadores, se escriben fuera del ciclo y al final, ya que el resultado se mostrará sólo una vez.

**Acumuladores**: Es una variable que almacena cantidades que cambian en cada iteración y se utiliza para efectuar sumas sucesivas. La principal diferencia con el contador es que el incremento o decremento de cada suma es variable en lugar de constante como en el caso del contador.

Acumulador = acumulador + variable

Imagine un acumulador como un canasto o carrito del mercado, en el cual, usted depositará algo cada vez que necesite o ejecute una acción, así como el carrito del mercado, dicho acumulador debe empezar en cero antes de iniciar el ciclo.

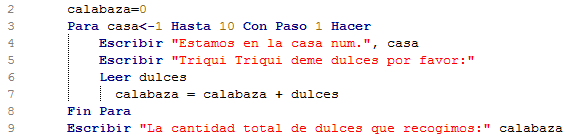
**Por ejemplo:**

Vamos a realizar un algoritmo para recoger dulces en Halloween, la acción “Triqui Triqui deme dulces” se ejecutará la cantidad de veces como casas vamos a visitar, por ejemplo, iremos a 10 casas, por lo que el ciclo se hará desde uno hasta 10.



|  |  |  |
| --- | --- | --- |
|  | Antes de empezar a pedir dulces, nuestra calabaza debe estar limpia y vacía, ese será nuestro acumulador, y antes de empezar el ciclo debe estar en cero:  La entrada de este algoritmo serán los dulces, que se almacenarán en la calabaza, y que se ingresarán a nuestro acumulador cuando estemos en cada casa. | |
|  | | La cantidad de dulces que contiene la calabaza lo conoceremos al final, después de recorrer las 10 casas, cuando contemos los dulces, por eso el resultado debe mostrarse por fuera del ciclo. |

**Código el PSeint:**



En la línea 2 la calabaza empieza vacía, antes de empezar el ciclo.

Línea 3, se tiene el ciclo desde la casa 1 hasta la 10.

En la línea 6 se ingresan los dulces que entregan en cada casa.

Línea 7, acumulador, en la calabaza se guardan los dulces de cada iteración.

En la línea 9 se muestran los dulces que se almacenaron en la calabaza al final del ciclo.

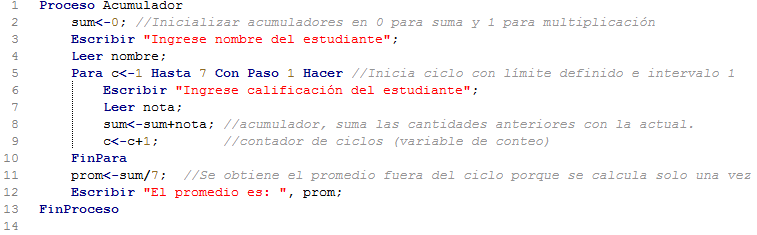
**Consulta**

Si quieres 0.5 adicionales en el segundo quiz, consulta el concepto y un ejemplo de Ciclo FOR Anidado, debes estar en capacidad de explicarlo.

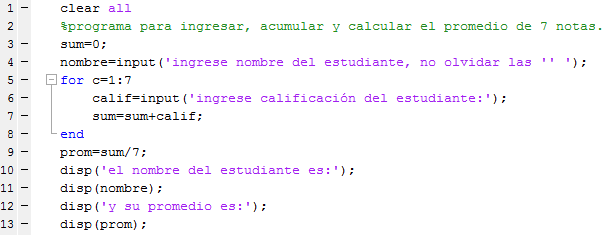
Será de utilidad para la solución del siguiente taller.

**TALLER**

1. Calcular el promedio de un estudiante que tiene 7 calificaciones en la materia de Diseño Estructurado de Algoritmos

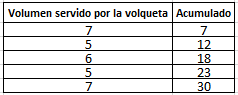


Realizar la prueba de escritorio para verificar el funcionamiento de este programa.

Código en MATLAB

1. Sumar los primeros N números.
2. Dado una cantidad límite de números, sumar todos los números que se ingresen por teclado hasta llegar a dicho límite.
3. Dado un límite N, sumar los números de forma consecutiva hasta dicho límite.
4. Calcular el factorial de un límite N ingresado.
5. Dado N calcular: (11)+ (22)+ (33)+... (NN)
6. 10 personas se reúnen para realizar una recolecta o vaca, ingresada la cantidad de dinero aportado por cada una, obtener el total recolectado.
7. Mostrar los números impares hasta in límite ingresado, también mostrar su acumulado.
8. A una fiesta asistieron N cantidad de personas de diferentes géneros, construir un algoritmo del cual se obtengan cuantos hombres y cuantas mujeres asistieron.
9. Se desea obtener el promedio de edades de un grupo de N estudiantes, contar cuantos son menores de edad.
10. El encargado de una obra recibe un suministro de volquetas con cemento. El control del volumen recibido lo lleva a través de una tabla donde apunta el volumen servido por la volqueta que llega y volumen acumulado a lo largo del día.

Por ejemplo:



El programa debe permitir el ingreso de la cantidad de veces que la volqueta va a descargar o servir, también debe permitir el ingreso del volumen que trae cada vez que descarga, y debe dar como resultado el acumulado, en cada descarga y en el total..

1. Leer 20 números y escribir cuantos son positivos, cuantos negativos y cuantos ceros.
2. Al cerrar un expendio de naranjas, 15 clientes que aún no han pagado recibirán un 15% de descuento si compran más de 10 kilos. Determinar cuánto pagara cada cliente y cuánto percibirá la tienda por esas compras.
3. Una persona desea invertir su dinero en un banco, el cual le otorga un 2% de interés. ¿Cuál será la cantidad de dinero que esta persona tendrá al cabo de un año si la ganancia de cada mes es reinvertida?
4. Escribir en programa que lea, una a una, 10 caracteres o letras. Luego indicar cuantas letras correspondientes a cada vocal se ingresaron, o sea, cuantas “a”, cuantas “e” y así sucesivamente.
5. En un supermercado un cajero captura los precios de los artículos que los clientes compran e indica a cada cliente cual es el monto de lo que deben pagar. Al final del día le indica a su supervisor cuanto fue lo que cobro en total a todos los clientes que pasaron por su caja. (se sugiere realizar este ejercicio con ciclos FOR anidados).
6. Elaborar un algoritmo que, dado un valor de **X** y un límite **n**, obtenga el valor de E mediante la suma de la serie: (tener en cuenta que la serien termina en el quinto término).

|  |  |
| --- | --- |
|  |  |

1. Desarrolle un algoritmo para una empresa que le permita calcular y mostrar el total de la nómina de sus 15 empleados, a cada uno de ellos se les cancela por horas trabajadas, y cada hora tiene un costo de $12.000.
2. Un Zoólogo pretende determinar el porcentaje de animales que hay en las siguientes tres categorías de edades: de 0 a 1 año, de más de 1 año y menos de 3 y de 3 o más años. El zoológico todavía no está seguro del animal que va a estudiar. Si se decide por elefantes solo tomara una muestra de 20 de ellos; si se decide por las jirafas, tomara 15 muestras, y si son chimpancés tomara 40.
3. Se necesita un programa para calcular resistencias series o paralelo, el usuario debe escoger mediante un menú cual configuración, y luego el programa le debe permitir ingresar cuántas resistencias va a calcular.
4. Un número perfecto es aquel que es igual a la suma de sus divisores:

6 es un número perfecto porque sus divisores son 1, 2 y 3; y 6 = 1 + 2 + 3.

Dado un número, averiguar si es perfecto, en cualquier caso, mostrar sus divisores.

1. Un número de Armstrong es todo aquel que es la suma de cada uno de sus mismos dígitos elevado al número total de dígitos.

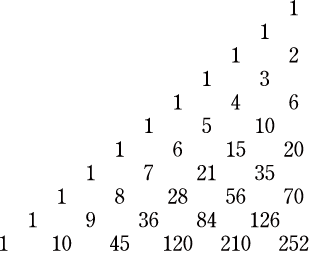
Por ejemplo el número 153 es de Armstrong ya que este posee 3 dígitos y la suma de cada uno elevado a 3 es igual a 1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153.

Dado un número, verificar si es un número Armstrong.

1. Calcular la serie de Fibonacci hasta un límite ingresado:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89… (Si utiliza vectores, se espera que esté en capacidad de explicarlo detenidamente).

1. Calcular la mitad del triángulo de Pascal ingresando el número de fila correspondiente.



**CICLO MIENTRAS (While)**

Esta es una estructura similar en algunas características, a la anterior, repite un proceso una cantidad de veces, pero a diferencia del “Para - hasta”, el “Mientras” lo hace mientras la condición inicial sea cierta, la cantidad de veces que el ciclo se ejecuta es indeterminada.

Por otra parte, esta estructura permite realizar el proceso cuando menos una vez, ya que se asume que la condición de entrada es cierta y se evalúa al final del proceso.

Inicio de condición

Mientras condición cierta

Accion1

Accion2

AccionN

Verificación de la condición

Fin mientras.

Existen algunos ejercicios con los cuales se pueden implementar algoritmos tanto con el “Para” (For), como con el “mientras” (While)

**Ejemplo 1:** imprimir los 10 primeros números:

N=1

Mientras N<=10 haga

Imprimir N

N=N+1

Fin mientras

**Ejemplo 2:** existen otras aplicaciones para el ciclo mientras:

n=1

mientras n=1 haga

imprimir “hola”

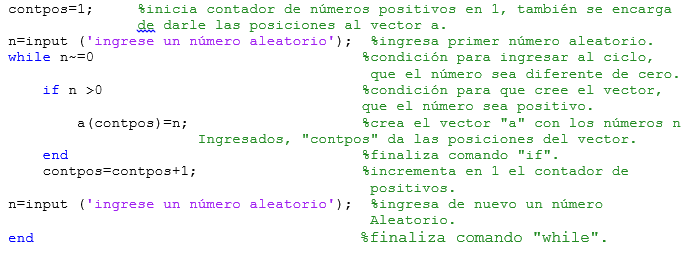
n=desea continuar? 1 para si 0 para no. % de esta manera el usuario elige cuando

terminar el programa.

Fin mientras

**Ejemplo 3: ciclo While en Matlab:**

Ingresar números aleatorios con un ciclo While, terminar la secuencia con un cero, y crear un vector con los números positivos ingresados en dicha secuencia.

Se sugiere la digitación del programa para ver su funcionamiento.

**Actividad**:

Desarrollar estos ejercicios con el ciclo mientras:

1. Probar los anteriores ejemplos por medio de una prueba de escritorio.
2. Leer una serie de números e imprimir su cuadrado (N2), terminar la secuencia con un número negativo.
3. De unos números que se leen por teclado, crear un vector con aquellos que son pares. terminar el programa con un 0.
4. Calcular y mostrar el producto de unos números distintos de cero.
5. inventar un algoritmo para calcular la edad de una persona, y que el usuario decida cuando salir.
6. Para una secuencia de números aleatorios, decir cuántos son ceros.
7. Realizar un algoritmo que permita ingresar peso, talla y nombre, y obtener como resultado el promedio de los pesos y de las tallas, el algoritmo se ejecuta una cantidad indeterminada de veces.