**III. Repeticiones**

Recuerden que los lenguajes de programación proveen estructuras que permiten repetir un bloque de

instrucciones una cantidad definida de veces. Las repeticiones pueden ser:

1. Exactas

a. Según un valor constante

b. Según una cantidad conocida a priori

2. No Exacta

a. Pre Condicional (0..N)

b. Pos Condicional (1..N)

**Ej. MI-12** Informar los primeros 100 números naturales y su sumatoria

Objetivo: Informar por pantalla los primero 100 número naturales y el resultado de sumarlos todos.

Datos de Entrada:

ninguna.

Precondiciones:

no hay.

Datos de salida:

sumatoria: Tipo Entero.

Ej. MI-13: Dados N y M números naturales, informar su producto por sumas sucesivas.

Objetivo: Mostrar el producto de dos números naturales dados mediante sumas sucesivas.

Datos de Entrada:

num1: Variable tipo Entero

num2: Variable tipo Entero

Precondiciones:

num1 >= 0

num2 >= 0

Datos de salida:

producto: Tipo Entero.

Flujo de Prueba:

|  |  |  |
| --- | --- | --- |
| Num1 | Num2 | producto |
| 2 | 1 | 2 |
| 0 | 3 | 0 |

Ej. MI-14: Dados 50 números enteros, informar el promedio de los mayores que 100 y la suma de los menores que

–10.

Objetivo: Calcular el promedio de números mayores a 100 y la suma de los menores que -10 de 50 números dados

Datos de Entrada:

num: Variable tipo Entero.

Precondiciones:

no hay.

Datos de salida:

promedioMayores: Tipo Real.

SumaMenores: Tipo Entero.

Flujo de Datos de Prueba:

|  |  |  |
| --- | --- | --- |
| num | promedioMayores | sumaMenores |
| -20, - 10, 50(x46), 100, 200 | 200 | -20 |
| -30, -20, 75(x46), 150, 200 | 175 | -50 |
| 20(x50) | 0 | 0 |
| -30(x50) | 0 | -1500 |
| 200(x25), 300(x25) | 250 | 0 |

Ej. MI-15: Se realiza una inspección en una fábrica de pinturas, y se detectaron 20 infracciones. De cada infracción se tomó nota de los siguientes datos:

- Tipo de Infracción (1, 2, 3, ó 4)

- Motivo de la infracción

- Valor de la multa

- Gravedad de la infracción (‘L’,‘M’, ‘G’)

Se pide informar al final del proceso:

• Los valores totales de la multa a pagar de acuerdo al tipo de gravedad.

• La leyenda “Clausurar fábrica” si la cantidad de infracciones 3 y 4 con gravedad “G” sean mayor a 3.

Objetivo: Determinar si se debe o no clausurar una fábrica de acuerdo a unas infracciones dadas.

Datos de Entrada:

tipoInfracion: Variable tipo Entero

motivoInfraccion: variable tipo Cadena

valorMulta: variable tipo Real.

gravedadInfraccion: Variable tipo Caracter

CANTIDAD\_INFRACCIONES: constante tipo Entero.

Precond:

tipoInfraccion = [1, 2, 3, 4]

valorMulta > 0

gravedadInfraccion = [‘L’,’M’,’G’]

CANTIDAD\_VALORES = 20

Datos de Salida:

mensaje: Tipo Cadena

totalL

totalM

totalG

Ej. MI-16: Ingresar e informar valores, mientras que el valor ingresado no sea negativo. Informar la cantidad de valores ingresados.

Objetivo: Mostrar la cantidad total de valores ingresados y cada valor ingresado.

Datos de entrada:

num: variable tipo Real

Precondiciones: ninguna

Datos de Salida:

num: tipo Real

cantValores: tipo Entero.

Ej. MI-17: Se ingresa un conjunto de valores float, cada uno de los cuales representan el sueldo de un empleado, excepto el último valor que es cero e indica el fin del conjunto. Se pide desarrollar un programa que determine e informe:

a) Cuántos empleados ganan menos $1.520.

b) Cuántos ganan $1.520 o más pero menos de $2.780.

c) Cuántos ganan $2.780 o más pero menos de $5.999.

d) Cuántos ganan $5.999 o más.

Objetivo:

Datos de Entrada:

sueldo: Variable tipo de Real.

Precond:

sueldo >= 0

Datos de Salida:

mensaje.

**Ej. MI-18:** Dado un valor M determinar y emitir un listado con los M primeros múltiplos de 3 que no lo sean de 5, dentro del conjunto de los números naturales.

Objetivo: Determinar una cantidad dada de primeros múltiplos de 3 que no sean múltiplo de 5.

Datos de Entrada:

nMultiplos Variable tipo Entero.

Precondiciones:

nMultiplos >= 0.

Datos de Salida:

mensaje: Tipo Cadena.

Ej. MI-19: Reescribir los algoritmos MI-16 al MI-18, utilizando ciclos 1-n

Ej. MI-20: Dados 10 valores informar el mayor

Objetivo: Informar el mayor de 10 valores ingresados

Datos de entrada:

num: variable tipo Real;

CANT\_VALORES: constante tipo Entero.

Precondiciones:

CANT\_VALORES = 10

Datos de Salida:

numMayor: tipo Real

Flujo de Prueba:

|  |  |
| --- | --- |
| Num | numMayor |
| 5; 1; -1,5; 50; -2; 35; 45; 15; -35; 4 | 50 |

Ej. MI-21: Dados N valores informar el mayor, el menor y en qué posición del conjunto fueron ingresados.

Objetivo: informar el mayor y menos de un conjunto de números dados y decir cuándo fueron ingresados.

Datos de entrada:

num: variable tipo Real

Precond: no hay

Datos de salida:

mayor

menor

Ej. MI-22: Dado un conjunto de Nombres y Fechas de nacimientos (AAAAMMDD), que finaliza con un Nombre = ‘FIN’, informar el nombre de la persona con mayor edad y el de la más joven.

Objetivo: Informar el nombre de la persona con mayor edad y el de la menor edad. Terminar el programa si nombre es igual a ‘FIN’;

Datos de Entrada:

nombre: variable tipo Cadena

fechaNac: vatiable tipo Entero

Precondiciones:

fechaNac >= 00000000 y fechaNac <= 99991231. Fecha válida en formato AAAAMMDD

Datos de salida:

mensaje: tipo Cadena

Ej. MI-23. Dado un conjunto de valores, que finaliza con un valor nulo, determinar e imprimir (si hubo valores):

a) El valor máximo negativo

b) El valor mínimo positivo

c) El valor mínimo dentro del rango -17.3 y 26.9

d) El promedio de todos los valores.

e)EN QUE POSICIÓN SE ENCONTRÓ EL MAYOR Y EL MENOR?

Objetivo: Imprimir la leyenda dada

Datos de Entrada:

num; variable tipo Real

Precondiciones:

Datos de Salida:

valorMaximoNeg: tipo Real

valorMinimoPos: tipo Real

minDelRango: tipo Real

promedio: tipo Real

posicionMayor: tipo Entero

posicionMenor: tipo Entero

Flujo de Datos de Prueba

Ej. MI-24: Se dispone de un lote de valores enteros positivos que finaliza con un número negativo.

El lote está dividido en sublotes por medio de valores cero. Desarrollar un programa que determine e informe:

a) por cada sublote el promedio de valores

b) el total de sublotes procesados

c) el valor máximo del conjunto, indicando en que sublote se encontró y la posición relativa del mismo dentro del sublote

d) valor mínimo de cada sublote

Nota: el lote puede estar vacío (primer valor negativo), o puede haber uno, varios o todos los sublotes vacíos (ceros consecutivos)

lote: negativo = lote vacío

lote positivo, negativo = 1 sublote con un valor

lote positivo, cero, positivo, negativo = 2 sublotes con un valor cada 1

lote +, +, 0 , +, +, 0, +, +, - = 3 sublotes, cada uno con 2 valores

Objetivo: Hacer un sistema que apile por sublotes hasta que se ingrese un número negativo. Cada sublote está separado por un cero. Informar

a) Promedio de valores por sublote,

b) Total de sublotes,

c) Valor máximo del lote, con número del sublote y posición dentro del sublote,

d) Valor mínimo de cada sublote

Datos de entrada:

num: variable tipo Real.

Precondiciones: ninguna

Datos de salida:

promedioValSublote: tipo Real

cantSublotes: tipo Entero

valorMaxDelLote: tipo Real

numSublote: tipo Entero

posicionSublote: tipo Entero

valorMinDelSublote: tipo Real

Ej. MI-25: Dada una serie de M pares {color, número} que corresponden a los tiros de una ruleta. Se pide informar:

a) cuántas veces salió el número cero y el número anterior a cada cero

b) cuántas veces seguidas llegó a repetirse el color negro

c) cuántas veces seguidas llegó a repetirse el mismo número y cuál fue

d) el mayor número de veces seguidas que salieron alternados el rojo y el negro

e) el mayor número de veces seguidas que se negó la segunda docenas

Objetivo: Ingresar una serie de pares {color, número} e informar.

Datos de entrada:

CANT\_PARES: constante tipo entero.

Color: variable tipo Cadena

num: variable tipo Entero

Precond:

CANT\_PARES >= 0

color = “rojo” o color = “negro”

num >= 0

Datos de Salida:

cantCeros: tipo Entero

anteriorAlCero: tipo Entero

cantRepNegro: tipo Entero

numRepetido: tipo Entero

cantRepNumero: Tipo Entero

mayorRepColorAlt: tipo Entero

mayorRepNeg2daDocena: tipo Entero

"Dado un conjunto de 10 números enteros, obtener el ranking de los 3 mayores"

Objetivo: Determinar el ranking de los 3 mayores número de 10 dados.

Datos de Entrada:

num: variable tipo Entero.

CANT\_NUMEROS: constante tipo Entero.

Precond:

CANT\_NUMEROS = 10

Datos de Salida:

mayor1: Tipo Entero

mayor2: Tipo Entero

mayor3: Tipo Entero