

Algoritmos y Programación

ARREGLOS

Repaso y ESTUDIO DE CASOS



Prof. Miguel A. Fernández



Algoritmos y Programación

Repasemos: Caso 1 – Almacenamiento de Datos

Se ingresan los siguientes datos de los 120 empleados de una empresa:

- Nro.de Legajo (rango de 1 a 120)
- Apellido y Nombre

Finalizado el ingreso de datos se debe:

- Imprimir un listado que contenga:

Nro.de Legajo – Apellido y Nombre

¿ Cuántos
arrays se
necesitan ?



Prof. Miguel A. Fernández



Algoritmos y Programación

Solución:

Program Empleados

Type
Nombres = array [1..120] of String;
Var
Nombre : Nombres;
LEG, i : Integer;

Inicio

Ingresar LEG
Mientras LEG <> 0
Ingresar Nombre(LEG)
Ingresar LEG
FinMientras

Para i = 1, 120, 1
Imprimir i, Nombre(i)
FinPara

FIN



Prof. Miguel A. Fernández



Algoritmos y Programación

Repasemos: Caso 2 – Array como Acumulador

Se ingresan los siguientes datos de los empleados de una empresa:

- Nro.de Legajo
- Apellido y Nombre
- Categoría (rango de 1 a 24)

Imprimir un listado con los datos ingresados.

Finalizado el ingreso de datos se debe:

- Imprimir un listado que contenga:

Categoría – Cantidad de Empleados

¿ Cuántos
arrays se
necesitan ?



Prof. Miguel A. Fernández



Algoritmos y Programación

Solución:

Program Empleados

Type
Categorias = array [1..24] of Integer;
Var
Cate : Categorias;
LEG, CAT, i : Integer;
NOM : String;

Inicio

Para i = 1, 24, 1
Cate(i) := 0
FinPara

Ingresar LEG
Mientras LEG <> 0
Ingresar NOM, CAT
Imprimir LEG, NOM, CAT
Cate(CAT) := Cate(CAT) + 1
Ingresar LEG
FinMientras

Para i = 1, 24, 1
Imprimir "Total Cat.: ", i, Cate(i)
FinPara

FIN



Prof. Miguel A. Fernández



Algoritmos y Programación

Volvamos al Caso 1: Ahora agreguemos un dato al proceso:

Se desea ingresar los siguientes datos de los 120 empleados de una empresa:

- Nro.de Legajo (rango de 1 a 120)
- Apellido y Nombre
- DNI

Finalizado el ingreso de datos se debe:

- Imprimir un listado que contenga:

Nro.de Legajo DNI Apellido y Nombre

¿ Cuántos
arrays se
necesitan ?



Prof. Miguel A. Fernández

Algoritmos y Programación

Arreglos a utilizar:

Array DNI:

12345	23456	34567	98765
1° Elemento	2° Elemento	3° Elemento	120° Elemento

Arrays Encadenados

DNI y Nombre del empleado MARÍA
DNI y Nombre del empleado JUAN

Prof. Miguel A. Fernández

Algoritmos y Programación

RAM

Array DNI:

Array Nombre:

Dado el código:

```
I := 2
Imprimir DNI(i), Nombre(i)
```

Variable i

23456 Juan

Prof. Miguel A. Fernández

Algoritmos y Programación

Solución:

Program Empleados

```
Type
DNIs = array [ 1..120] of integer;
Nombres = array [1..120] of string;
Var
DNI : DNIs;
Nombre : Nombres;
LEG, i : Integer;
```

```
Inicio
  Ingresar LEG
  Mientras LEG <> 0
    Ingresar DNI(LEG), Nombre(LEG)
    Ingresar LEG
  FinMientras
  Para i = 1, 120, 1
    Imprimir DNI(i), Nombre(i)
  FinPara
FIN
```

Prof. Miguel A. Fernández

Algoritmos y Programación

Ahora agreguemos un dato mas al proceso:

Se desea ingresar los siguientes datos de los 120 empleados de una empresa:

- Nro.de Legajo (rango de 1 a 120)
- Apellido y Nombre
- DNI
- Dirección

Finalizado el ingreso de datos se debe:

- Imprimir un listado que contenga:

Nro.de Legajo - DNI - Apellido y Nombre - Dirección

¿ Cuántos arrays se necesitan ?

Prof. Miguel A. Fernández

Algoritmos y Programación

Solución:

Program Empleados

```
Type
DNIs = array [ 1..120] of integer;
Datos = array [1..120] of string;
Var
DNI : DNIs;
Nombre : Datos;
Direccion : Datos;
LEG : Integer;
i : Integer;
```

```
Inicio
  Ingresar LEG
  Mientras LEG <> 0
    Ingresar DNI(LEG), Nombre(LEG), Dirección(LEG)
    Ingresar LEG
  FinMientras
  Para i = 1, 120, 1
    Imprimir DNI(i), Nombre(i), Dirección(i)
  FinPara
FIN
```

Prof. Miguel A. Fernández

Algoritmos y Programación

Caso: Dos arrays. Uno para acumular:

De cada venta que ha realizado una casa de venta de tractores durante todo un mes, se ingresa por teclado:

- Fecha
- Nro. de Vendedor (rango de 1 a 25)
- Importe de la venta

Se requiere imprimir un listado con los datos ingresados.

Finalizado el ingreso de datos se debe imprimir un listado que contenga:

Nro.de Vendedor - Apellido y Nombre - Importe total vendido

Nota: El nombre de cada vendedor se debe ingresar al inicio del código.

Prof. Miguel A. Fernández

Ejercicio Básico

Solución:

Program Tractor

```

Type
  Totales = array [ 1..25] of Real;
  Nombres = array [1..25] of string;
Var
  Total : Totales;
  Nombre : Nombres;
  FEC, NUV, IMP, i : Integer;

```

```

Inicio
  Ingresar NUV
  Mientras NUV <> 0
    Ingresar Nombre(NUV)
    Total(NUV) := 0
    Ingresar NUV
  FinMientras

  Ingresar FEC
  Mientras FEC <> 0
    Ingresar NUV, IMP
    Imprimir FEC, NUV, IMP
    Total(NUV) := Total(NUV) + IMP
    Ingresar FEC
  FinMientras

  Para i = 1, 25, 1
    Imprimir i, Nombre(i), Total(i)
  FinPara
FIN

```

Prof. Miguel A. Fernández



Algoritmos y Programación

Caso: Búsqueda en arreglos:

Para procesar los datos de las obras en construcción de la provincia se debe:

- Ingresar los siguientes datos de los arquitectos directores de obra:
 - Nro. de matrícula (números NO correlativos)
 - Apellido y nombres del arquitecto.
- Codificar un módulo de consulta tal que ingresando el número de matrícula se muestre por pantalla el Apellido y nombre del arquitecto.

Nota: Existen 75 arquitectos habilitados para dirigir obras.



Prof. Miguel A. Fernández

Algoritmos y Programación

TIPOS DE BÚSQUEDA DE UN ELEMENTO EN ARREGLOS

DIRECTA

SE CONOCE
SU
UBICACIÓN

Se accede al elemento
utilizando un subíndice
inicializado con la posición
que ocupa el elemento.

SECUENCIAL

NO SE CONOCE SU
UBICACIÓN

Se recorre el arreglo, elemento
por elemento hasta encontrar el
dato buscado o hasta llegar al
final del arreglo.

Prof. Miguel A. Fernández

Ejercicio Básico

Solución:

Program OBRAS

```

Type
  Matris = array [ 1..75] of integer;
  Nombres = array [1..75] of string;
Var
  Matri : Matris;
  Nombre : Nombres;
  NUM, i : Integer;
  ESTA : Boolean;

```

```

Inicio
  Para i = 1, 75, 1
    Ingresar Matri(i), Nombre(i)
  FinPara

  Ingresar NUM
  Mientras NUM <> 0
    ESTA := False
    i := 0
    Mientras i < 75 and ESTA = False
      i := i + 1
      Si Matri(i) = NUM
        Mostrar Nombre(i)
        ESTA := True
      FinSi
    FinMientras
    Si ESTA = False
      Mostrar "No existe Matrícula"
    FinSi
    Ingresar NUM
  FinMientras
FIN

```



Prof. Miguel A. Fernández

Algoritmos y Programación

Caso: Arreglos con dimensión desconocida:

Para procesar los datos de las obras en construcción de la provincia se debe:

- Ingresar los siguientes datos de los arquitectos directores de obra:
 - Nro. de matrícula (números NO correlativos)
 - Apellido y nombres del arquitecto.
- Codificar un módulo de consulta tal que ingresando el número de matrícula se muestre por pantalla el Apellido y nombre del arquitecto.



Prof. Miguel A. Fernández



Algoritmos y Programación

Estrategia:

- Definir un arreglo con una dimensión suficientemente grande.
- Ingresar los datos y almacenar la posición del último elemento con dato en una variable.
- Para todo el proceso posterior, considerar la dimensión del arreglo desde el inicio hasta el valor de la variable.

Ejemplo:

```

Type
  Vector = array [1..16] of integer;
Var
  Vec : Vector;
  i, z, Dato : Integer;

```

```

Inicio
  i := 0
  Ingresar DATO
  Mientras DATO <> 0
    i := i + 1
    VEC(i) := DATO
    Ingresar DATO
  FinMientras
  z := i

```

```

Para i = 1, z, 1
  Imprimir Vec(i)
FinPara

```

Posición: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16



Ejercicio Básico

Program Obras

Type

```
Matris = array [ 1..500] of integer;
Nombres = array [1..500] of string;
Var
  Matri : Matris;
  Nombre : Nombres;
  NUM, i, z : Integer;
  ESTA : Boolean;
```

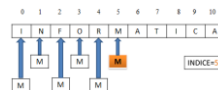
Inicio

```
Ingresa NUM
i := 0
Mientras NUM <> 0
  i := i + 1
  Matri(i) := NUM
  Ingresa Nombres(i)
  Ingresa NUM
FinMientras
```

```
z := i
Ingresa NUM
Mientras NUM <> 0
  ESTA := False
  i := 0
  Mientras i < z and ESTA = False
    i := i + 1
    Si Matri(i) = NUM
      Mostrar Nombre(i)
      ESTA := True
    FinSi
  FinMientras
  Si ESTA = False
    Mostrar "No existe Matricula"
  FinSi
  Ingresa NUM
FinMientras
FIN
```

Algoritmos y Programación

RESUMEN:



Hemos visto:

- Arreglos encadenados
- Búsqueda secuencial
- Arreglos con dimensión desconocida

Algoritmos y Programación

¡ TIEMPO DE TRABAJO !



Algoritmos y Programación

Una empresa debe procesar los días de licencia que le corresponde a sus operarios.

Se requiere:

- Ingresa de cada operario los días de licencia que le corresponde:
 - Nro. de Documento
 - Cantidad de días de licencia anual que le corresponde
 - Procesar novedades. Cada vez que un operario solicita tomar días licencia se ingresa:
 - Nro. de Documento
 - Cantidad de días solicitados
- Se debe:
- Verificar que tenga la cantidad de días disponibles. Si no los tiene rechazar la operación.
 - Si tiene días disponibles, restar los días solicitados de la cantidad disponible.
- c) Al final del proceso imprimir un listado que contenga:

Número de Documento – Cantidad de días disponibles

Cuando finaliza el listado imprimir:

- Cantidad de operarios que tiene la empresa
- Nro. de Documento del operario que tiene mayor cantidad de días disponibles.

Nota: Se trata de una empresa mediana de no mas de 250 operarios.

Ejercicio

Program Tarjetas

Type

```
Comun = array [1..250] of Integer;
Var
  Docu : Comun;
  Dias : Comun;
  DOC, MDOC, DIA, MAX : Integer;
  ESTA : Boolean;
```

Begin

```
i := 0
Ingresa DOC
Mientras DOC <> 0
  i := i + 1
  Docu(i) := DOC
  Ingresa Dias(i)
  Ingresa DOC
FinMientras
z := i
```

```
Ingresa DOC
Mientras DOC <> 0
  Ingresa DIA
  i := 0
  ESTA := False
  Mientras i < z and ESTA = False
    i := i + 1
    Si Docu(i) = DOC
      Si Dias(i) >= DIA
        Dias(i) := Dias(i) - DIA
      Sino
        Mostrar "Días Insuficiente"
      FinSi
      ESTA := True
    FinSi
  FinMientras
  Si ESTA = False
    Mostrar "Documento no Encontrado"
  FinSi
  Ingresa DOC
FinMientras
```

Ejercicio

Rutina Impresión

```
MAX := Min-Valor
Para i = 1, z, 1
  Imprimir Docu(i), Dias(i)
  Si Dias(i) > MAX
    MAX := Dias(i)
    MDOC := Docu(i)
  FinSi
FinPara
Imprimir "Cantidad Operarios: ", z
Imprimir "Doc. Operario con > Días: ", MDOC
FIN
```

Algoritmos y Programación

Una tarjeta de créditos asigna puntajes por cada compra que realizan sus 1.250 clientes. Al final del año los puntos obtenidos el cliente lo puede canjear por diversos productos.

Se requiere:

a) Ingresar los siguientes datos de los clientes de la tarjeta:

- Nro. De Tarjeta (8 dígitos **NO correlativos**)
- Apellido y nombres del cliente.

b) Procesar las compras y sumar los puntos que correspondan a cada cliente.

De cada compra se ingresa por teclado:

- Nro. de Tarjeta
- Importe de la compra

Se debe sumar al cliente: 150 puntos si el importe de la compra es menor a \$ 10.000.- de lo contrario sumar 220 puntos.

Por cada compra imprimir:

Apellido y Nombre del cliente - Importe Compra - Puntos sumados

c) Al final del proceso imprimir un listado que contenga:

Apellido y nombre del cliente -- Total de puntos acumulados

