



# Conceptos de Algoritmos Datos y Programas

# CADP – TEMAS



- Estructura de datos ARREGLO
- Agregar elementos
- Insertar elementos
- Eliminar elementos

Carga de valores

Lectura / Escritura

Recorridos

Dimensión física y lógica

Agregar elementos

Insertar elementos

Borrar elementos

Búsqueda de un elemento





Significa agregar en el vector un elemento detrás del último elemento cargado en el vector. Puede pasar que esta operación no se pueda realizar si el vector está lleno

D1 = 4

45

a

34	10	-1	5						
1	2	3	4	5	6	7	8	9	10

Qué pasos  
considero?



Significa agregar en el vector un elemento detrás del último elemento cargado en el vector. Puede pasar que esta operación no se pueda realizar si el vector está lleno

- 1- Verificar si hay espacio (cantidad de elementos actuales es menor a la cantidad de elementos posibles)
- 2- Agregar al final de los elementos ya existentes el elemento nuevo.
- 3- Incrementar la cantidad de elementos actuales.

**Cómo se  
implementa?**



Dado un vector de números enteros (10 elementos como máximo) realice un programa que lea un nuevo número e invoque a un módulo que agregue el elemento en el vector.

```
Program uno;  
  const  
    fisica = 10;  
  type  
    numeros= array [1..fisica] of integer;
```

VN

?	?	?	?	?	?	?	?	?	?
---	---	---	---	---	---	---	---	---	---

```
var  
  VN: numeros;  
  dimL, valor:integer;  
  ok:boolean;
```

dimL = 4

VN

4	-1	10	3	?	?	?	?	?	?
---	----	----	---	---	---	---	---	---	---

```
Begin  
  cargar (VN,dimL);  
  read(valor);  
  agregar(VN,dimL,ok,valor);
```

valor = 7      dimL = 5      ok = true

VN

4	-1	10	3	7	?	?	?	?	?
---	----	----	---	---	---	---	---	---	---

End.

```
Procedure agregar (var a :números; var dL:integer; var pude:boolean; num:integer);
```

```
Begin
```

```
  pude:= false;
```

**Verifico si hay espacio**

```
  if ((dL + 1) <= física) then
```

```
    begin
```

```
      pude:= true;
```

```
      dL:= dL + 1;
```

```
      a[dL]:= num;
```

```
    end;
```

```
end.
```

**Registro que se pudo realizar  
Incremento la dimensión lógica  
Agrego elelemento**



Significa agregar en el vector un elemento en una posición determinada. Puede pasar que esta operación no se pueda realizar si el vector está lleno o si la posición no es válida

**D1 = 4**

**45**

**pos = 2**

**a**

34	10	-1	5						
1	2	3	4	5	6	7	8	9	10

Qué pasos  
considero?





Significa agregar en el vector un elemento en una posición determinada. Puede pasar que esta operación no se pueda realizar si el vector está lleno o si la posición no es válida

- 1- Verificar si hay espacio (cantidad de elementos actuales es menor a la cantidad de elementos posibles)
- 2- Verificar que la posición sea válida (esté entre los valores de dimensión definida del vector y la dimensión lógica).
- 3- Hacer lugar para poder insertar el elemento.
- 4- Incrementar la cantidad de elementos actuales.

**Cómo se  
implementa?**



Dado un vector de números enteros (10 elementos como máximo) realice un programa que lea un nuevo número y una posición e invoque a un módulo que inserte el elemento en el vector en la posición leída.

```
Program uno;
  const
    fisica = 10;
  type
    numeros= array [1..fisica] of integer;
```

VN

?	?	?	?	?	?	?	?	?	?
---	---	---	---	---	---	---	---	---	---

dimL = 4

```
var
  VN: numeros;
  dimL, valor,pos:integer;
  ok:boolean;
```

VN

4	-1	10	3	?	?	?	?	?	?
---	----	----	---	---	---	---	---	---	---

```
Begin
  cargar (VN,dimL);
  read(valor); read(pos);
  insertar(VN,dimL,valor,ok,pos);
```

valor = 7      pos= 2      dimL = 5      ok = true

VN

4	7	-1	10	3	?	?	?	?	?
---	---	----	----	---	---	---	---	---	---

```
Procedure insertar (var a :números; var dL:integer; var pude:boolean;  
Var  
    num:integer; pos: integer);  
    i:integer;
```

```
Begin  
    pude:= false;  
  
    if ((dL + 1) <= física) and (pos>= 1) and (pos <= dL) )then begin  
  
        for i:= dL downto pos do  
            a[i+1]:= a[i];  
  
        pude:= true;  
        a[pos]:= num;  
        dL:= dL + 1;  
    end;  
end;
```

**Verifico si hay espacio y si la posición es válida**

**Corro los elementos empezando desde atrás hasta la posición a insertar para hacer el hueco donde se va a insertar el elemento**

**Registro que se pudo realizar  
Inserto el elemento  
Incremento la dimensión lógica**



Significa borrar (lógicamente) en el vector un elemento en una posición determinada, o un valor determinado. Puede pasar que esta operación no se pueda realizar si la posición no es válida, o en el caso de eliminar un elemento si el mismo no está

pos = 2	34	10	-1	5						
D1 = 4	1	2	3	4	5	6	7	8	9	10

a

Qué pasos  
considero?



Significa borrar (lógicamente) en el vector un elemento en una posición determinada, o un valor determinado. Puede pasar que esta operación no se pueda realizar si la posición no es válida, o en el caso de eliminar un elemento si el mismo no está

- 1- Verificar que la posición sea válida (esté entre los valores de dimensión definida del vector y la dimensión lógica).
- 2- Hacer el corrimiento a partir de la posición y hasta el final.
- 3- Decrementar la cantidad de elementos actuales

**Cómo se  
implementa?**



Dado un vector de números enteros (10 elementos como máximo) realice un programa que lea una posición e invoque a un módulo que elimine el elemento en el vector en la posición leída.

```
Program uno;
  const
    fisica = 10;
  type
    numeros= array [1..fisica] of integer;
```

VN

?	?	?	?	?	?	?	?	?	?
---	---	---	---	---	---	---	---	---	---

dimL = 4

```
var
  VN: numeros;
  dimL,pos:integer;
  ok:boolean;
```

VN

4	-1	10	3	?	?	?	?	?	?
---	----	----	---	---	---	---	---	---	---

```
Begin
  cargar (VN,dimL);
  read(pos);
  eliminar(VN,dimL,ok,pos);
```

pos= 2   dimL = 3   ok = true

VN

4	10	3	3	?	?	?	?	?	?
---	----	---	---	---	---	---	---	---	---

```
Procedure eliminar (var a :números; var dL:integer; var pude:boolean;pos: integer);
```

```
Var
```

```
  i:integer;
```

```
Begin
```

```
  pude:= false;      Verifico si la posición es válida
```

```
  if ((pos>= 1) and (pos <= dL) )then begin
```

```
    for i:= pos to (dL-1) do  
      a[i]:= a[i+1];
```

**Corro los elementos empezando desde la posición hasta la dimensión lógica-1 para “tapar” el elemento a eliminar**

```
    pude:= true;  
    dL:= dL - 1;  
  end;
```

**Registro que se pudo realizar  
Decremento la dimensión lógica**

```
end;
```