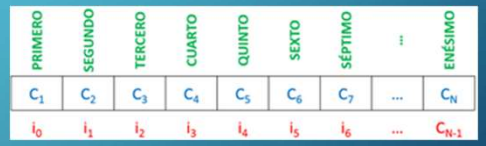
UTEM

FING
FACULTAD DE INGENIERÍA

ALGORITMOS Y PROGRAMACIÓN INFB8021

4 ARREGLOS



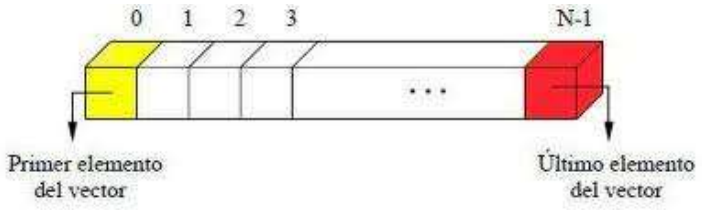
UTEM | FING
FACULTAD DE INGENIERÍA
Algoritmos y Programación

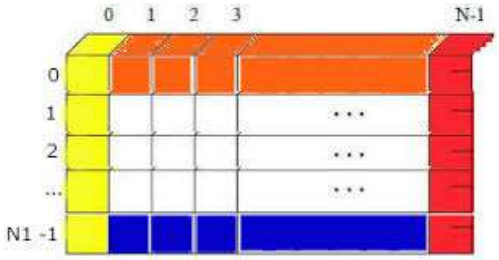
ARREGLOS

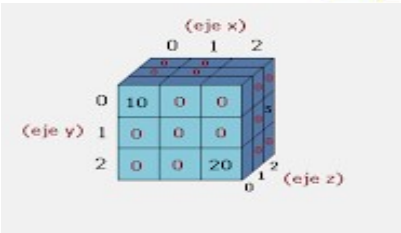
Introducción

Los arreglos, es un conjunto o tipo de variable estructurada, compuesto por una colección finita de datos del mismo tipo, a las cuales se puede acceder mediante un nombre ligado a un índice.

Un arreglo puede tener una o más dimensiones, pudiéndose diferenciar según la cantidad de dimensiones unidimensional, bidimensional, tridimensional,…, kdimensional. También es posible encontrar esta misma definición como arreglo unidimensional o Vector, arreglo bidimensional o Matriz, arreglo tridimensional o Cubo, toda vez que representan lo mismo.







UTEM | FING FACULTAD DE INGENIERÍA Algoritmos y Programación

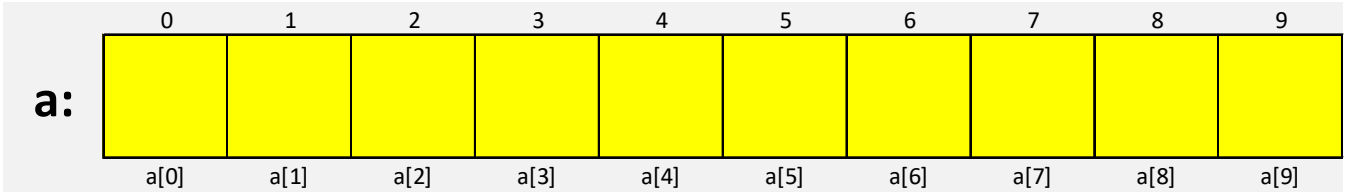
ARREGLOS

Estructura Base

La cantidad de dimensiones de un arreglo se representa mediante una cantidad igual de índices, El o los índice(s) permite(n) acceder directamente a un elemento específico del arreglo.

El valor 0 de un índice representa el primer elemento de la respectiva dimensión y los elementos del arreglo se almacenan en direcciones de memoria contiguas.

Ejemplo: Si cada elemento de un arreglo se referencia mediante un índice, entonces se trata de un arreglo unidimensional.
int a[10];



Declaración de Arreglos según dimensiones

```
int Vector [F];          -> Unidimensional
int Matriz [F][C];       -> Bidimensional
int Cubo   [X][Y][Z];    -> Tridimensional
```

UTEM | FING FACULTAD DE INGENIERÍA Algoritmos y Programación

ARREGLOS

Uso de typedef

typedef es una palabra reservada, su función es asignar un nombre alternativo a tipos existentes, Se recomienda definir un tipo de dato typedef el fin de proveer un modelo destinado a declarar variables de esta naturaleza en diferentes funciones.

Así por ejemplo podremos definir un nuevo tipo de dato que luego utilizamos para declarar nuestras variables

```
#include <iostream>

typedef int vector[100];
typedef int matriz[10][20];
typedef int cubo[5][8][5];

int main()
{
   vector v;
   matriz m;
   cubo c;
   return 0;
}
```

UTEM | FING FACULTAD DE INGENIERÍA Algoritmos y Programación

EJEMPLO

Implementar las siguientes funciones:

Inicializa Vector, que inicializa los valores de un Vector en 0.

Llena Vector Serie, que llena el vector con datos incrementados según una serie fija

```
void inicializavector (vector v, int n)
{
  int i;
  for (i=0;i<n;i++)
    v[i]=0;
}
```

```
a[0] = 0
a[1] = 0
a[2] = 0
a[3] = 0
a[4] = 0
---------
```

```
void llenarvectorserie (vector v, int s, int n)
{
  int i;
  int serie = s;
  for (i=0;i<n;i++)
  {
     v[i]=serie;
     serie=serie+s;
  }
}
```

```
a[0] = 3
a[1] = 6
a[2] = 9
a[3] = 12
a[4] = 15
----------
```

UTEM | FING FACULTAD DE INGENIERÍA Algoritmos y Programación

EJEMPLO

Implementar las siguientes funciones:

- Invierte Vector, que completa un segundo vector con los datos del primero pero en orden inverso.
- Imprime Vector, que imprime el contenido de un vector

```
void inviertevector (vector v1, vector v2, int n)
{
  int i=0;
  for (i=n-1;i>=0;i--)
    v2[i]=v1[n-i-1];
}
```

```
------------
b[0] = 15
b[1] = 12
b[2] = 9
b[3] = 6
b[4] = 3
```

```
void imprimevector (vector v, int n, char nombre)
{
  int i;
  for (i=0;i<n;i++)
    printf("%c[%d] = %d \n",nombre,i,v[i] );

}
```

```
a[0] = 3
a[1] = 6
a[2] = 9
a[3] = 12
a[4] = 15
-----------
```

UTEM | FING FACULTAD DE INGENIERÍA Algoritmos y Programación

EJEMPLO

```
#include <iostream>
typedef int vector[5];

void inicializavector (vector v, int n)
void llenarvectorserie (vector v, int s, int n)
void inviertevector (vector v1, vector v2, int n)
void imprimevector (vector v, int n, char nombre)

int main()
{
     vector a,b;
     int largo = 5;
     inicializavector(a,largo);
     imprimevector(a,largo,'a');
     llenarvectorserie(a,3,largo);
     cout << "-----------------" <<endl;
     imprimevector(a,largo,'a');
     inviertevector(a,b,largo);
     cout << "-----------------" <<endl;
     imprimevector(b,largo,'b');
     return 0;
}
```

```
> clang-7 -pthread -lm -o main main.c
> ./main
a[0] = 0
a[1] = 0
a[2] = 0
a[3] = 0
a[4] = 0
---------------------------
a[0] = 3
a[1] = 6
a[2] = 9
a[3] = 12
a[4] = 15
---------------------------
b[0] = 15
b[1] = 12
b[2] = 9
b[3] = 6
b[4] = 3
>
```