

Hoy les presentamos



LOS
SUPER VECTORES Y
MATRICES

¿Que es un vector?

En Informática se llama vector/arreglo a una colección de valores que alberga un conjunto de datos del mismo tipo ubicados cada uno, en una posición determinada y en una sola variable.

Los vectores los vamos a pensar e imaginar de la siguiente manera:



--	--	--	--	--	--	--

¿Como se puedo crear este tipo de variable?

tipo_de_dato nombre_vector[TAMANIO_VECTOR];

Ejemplo: unsigned int coleccion_libros[7];

TAMANIO_VECTOR será el tamaño que tendrá mi vector este tamaño debe ser constante y ser conocido a la hora de compilar.

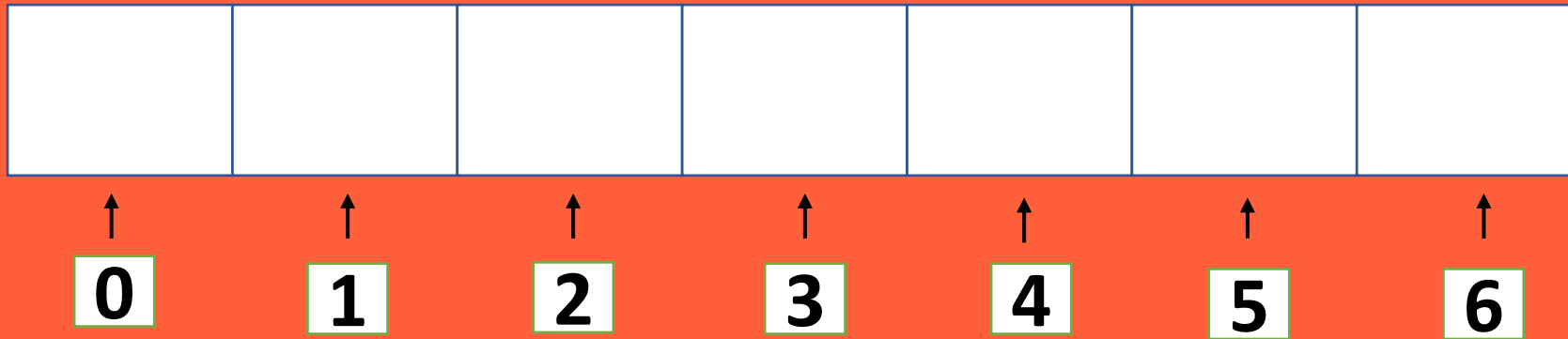
¿Por que no se puede cambiar el TAMAÑO?

En C cambiar el tamaño de un vector no es posible, por lo cual debemos buscar otra forma de realizarlo si necesitamos en algun momento incrementar nuestra colección, para esto lo que vamos a hacer es crear un vector de tamaño muy grande (la cota superior de nuestro problema Que será constante) y vamos a tener una variable entera (la llamaremos tope la cual va a variar) que nos dice la cantidad de elementos que tiene nuestro vector.



¿Como Podemos acceder a cada elemento que contiene el vector?

Para acceder a cada elemento tendremos que tener en cuenta que los vectores cuentan con indices que comienzan desde el cero ¿Como es esto?



Esto significa que para cualquier operación vamos a necesitar el vector, su tamaño y tope, siempre van de la mano. Como el tope es algo que creamos nosotros, tendremos que actualizarlo constantemente.



¿Que operaciones podemos hacer con vectores?



- ❖ Los vectores no tienen operaciones por defecto (pero podemos crearlas nosotros).
- ❖ Cosas como “ $\text{vector}_1 + \text{vector}_2$ ” no se pueden hacer
- ❖ Cualquier operación que queramos hacer entre vectores tenemos que hacerla iterando elemento a elemento.

Desde que nacieron las chicas super poderosas Bombón, Burbuja y Bellota guardaron la cantidad de regalos que les dieron hasta al momento por cada cumpleaños, incluyendo el día que nacieron. Es decir,
al momento de nacer 0 años \rightarrow 3 regalos, 1 año \rightarrow 4 regalos....
En la actualidad tienen 8 años y esperan recibir muchos mas.
En esta oportunidad las chicas piden ayuda a los alumnos de algo1Mendez para que realicen un programa el cual les diga cuantos regalos tienen hasta el momento sabiendo que tienen 8 años.




```
#include <stdio.h>
```

```
int main(){
```

```
    int regalos_cumple_cero=3;  
    int regalos_cumple_uno=4;  
    int regalos_cumple_dos=5;  
    int regalos_cumple_tres=6;  
    int regalos_cumple_cuatro=7;  
    int regalos_cumple_cinco=8;  
    int regalos_cumple_seis=9;  
    int regalos_cumple_siete=10;  
    int regalos_cumple_ocho=11;
```

```
    int total_regalos=0;  
    total_regalos= regalos_cumple_cero + regalos_cumple_uno + regalos_cumple_dos + regalos_cumple_tres +  
        regalos_cumple_cuatro + regalos_cumple_cinco + regalos_cumple_seis + regalos_cumple_siete +  
        regalos_cumple_ocho;
```

```
    printf("La cantidad de regalos que recibieron las chicas hasta ahora es: %i \n", total_regalos);
```

```
    return 0;
```

```
}
```

Mmm... Espero
que no usen
vectores!



Lo que vamos a pensar es tengo muchas variables del mismo tipo entonces las puedo agrupar en una variable.



TOPE




```

#include <stdio.h>

const int MAX_CUMPLES= 100;
const int REGALOS_INICIO=3;

/*
Pre condiciones: recibe un vector de regalos, su tope por referencia previamente inicializado y una edad menor a MAX_CUMPLES
Post condiciones: inicializa el vector con la cantidad de regalos correspondientes a la edad de las chicas.
*/
void inicializar_regalos(int regalos_cumple[MAX_CUMPLES], int* tope, int edad){
    for(int i=0; i<=edad; i++){
        regalos_cumple[i]= REGALOS_INICIO +i;
        (*tope)++;
    }
}

/*
Pre condiciones: recibe un vector inicializado con su correspondiente tope.
Post condiciones: devuelve la cantidad total de regalos que tienen las chicas hasta el momento.
*/
int total_regalos(int regalos_cumple[MAX_CUMPLES], int tope){
    int total_regalos=0;
    for(int i=0; i<tope; i++){
        total_regalos= total_regalos + regalos_cumple[i];
    }

    return total_regalos;
}

int main(){
    int regalos_cumple[MAX_CUMPLES];
    int tope=0;
    int edad=8;
    inicializar_regalos(regalos_cumple, &tope, edad);
    int regalos_totales= total_regalos(regalos_cumple, tope);

    printf("La cantidad de regalos que tienen las chicas hasta ahora son: %i \n", regalos_totales);

    return 0;
}

```



Matrices

- ❖ Las matrices son arreglos de dos dimensiones.
- ❖ Se comportan igual que los vectores pero con algunos cambios de sintaxis.

- Ejemplo:

- `tipo_de_dato nombre_matriz[CANT_FILAS][CANT_COLUMNAS]`

¿Existen Arreglos N-dimensionales?

Si existe, pero por lo general, cualquier problema puede ser resuelto con un arreglo de una dimensión pero a veces es más cómodo trabajar con más dimensiones.

En esos casos se pueden crear arreglos con más dimensiones. Para crearlos y usarlos la sintaxis es la misma que la matriz pero agregando más corchetes.



Veamos un ejemplo...