

# Patrones de resolución con vectores

En este documento comenzamos a trabajar con patrones de resolución de algoritmos, utilizando las estructuras de datos arreglos y registros.

## Temas a desarrollar

- Estructura de datos: matrices
- Aplicaciones con matrices
- Direcccionamiento indirecto

# Antes de comenzar con la ejercitación hagamos un resumen

Las estructuras o registros permiten agrupar datos de diversos tipos asociados a una entidad en particular. Por ejemplo las personas tienen documento, nombre, apellido, fecha de nacimiento, etc. Es muy útil para una buena comprensión del código y permite el pasaje de parámetros en forma clara y ordenada.

Los arreglos son estructuras de datos homogéneas (todos sus datos son del mismo tipo) que permiten almacenar un determinado número de datos bajo un mismo identificador, para luego referirse a los mismos utilizando uno o más subíndices. Los arreglos pueden pensarse como vectores, matrices, etc. Para poder utilizar un arreglo, primero es obligatorio su dimensionamiento; es decir, definirlo declarando los rangos de sus subíndices, lo cual determina cuántos elementos se almacenarán y cómo se accederá a los mismos. Todas las celdas de un arreglo son del mismo tipo, este puede ser un tipo simple (int, float, string, char, etc), un tipo complejo definido por el desarrollador (otro arreglo, una estructura, etc)

Hay que tener en cuenta que se utilizan arreglos, de 1 o 2 dimensiones, toda vez que se debe mantener en memoria varios datos de una misma característica, y se accede a los mismos a través de un índice. El índice varía entre 0 y la dimensión del arreglo menos 1.

**Los arreglos son estructuras estáticas, por lo tanto se debe definir durante el desarrollo del programa.**

## Y las matrices, ¿qué son?

Parece una pregunta rara...y lo es. ¿qué son?, o mejor dicho ¿para qué sirven?

Seguramente nos preguntamos lo mismo acerca de los vectores, ¿qué son?, ¿para qué sirven?, o mejor aún ¿cuándo se utilizan?. Entonces nos podemos parar en esta pregunta **¿cuándo se utilizan la matrices?**

**Recordemos**: un vector es una colección de elementos todos del mismo tipo y para un mismo propósito. Entonces, si se debe procesar un conjunto de datos homogéneos y realizar varias acciones sobre estos, entonces los vectores son las variables adecuadas para este propósito. Son fáciles de definir, su uso es muy sencillo, requieren poco código debido a que las soluciones son generalizables.

Claro, si muchas variables tienen el mismo nombre y se accede a cada una a través de un índice, entonces basta un proceso repetitivo (4 o 5 líneas) para recorrerlo y mostrar su contenido en la pantalla. ¿Y para ordenarlo?, también poco código (aunque los métodos son difíciles), pues se trata de recorridos, preguntas e intercambios entre variables con el mismo nombre y subindicadas. **BINGO**, es un **GOLAZO** trabajar con vectores

Pero entonces, ¿qué son las matrices?, o mejor dicho, ¿cómo se las puede aplicar? .

Pensemos, o mejor veamos un ejemplo. Estamos en un curso de AyED digamos el 1 (ejemplito), somos máximo 60, podemos meter (los datos) en un vector y hacer un montón de proceso.

Al lado está el curso 2, también máximo 60. Y a su costado está el 3, y lo sigue el 4, y así hasta el curso...10.

Entonces tenemos un vector por cada curso, todos del mismo tipo, y a los cuales se le va a realizar el mismo tratamiento. OH!!! (como diría la abuela), podemos pensar un vector de vectores.

**AHÍ TENEMOS UNA MATRIZ Y UNA APLICACIÓN**

## Comencemos a definir una matriz (arreglo de más de una dimensión)

Así las cosas , comencemos a definir una matriz. O mejor, intentemos definir (en C++) el ejemplo de la filmina anterior.

Primero hablamos de un estructura de datos de estudiantes. Luego mencionamos una colección de estudiantes (60) de un curso. Finalmente dijimos que hay varios cursos (10) que tienen estudiantes (la misma cantidad máxima) con el mismo tipo de datos.

Vamos, escribamos todo esto en código...

```
#define _CANTEST 60
#define _CANTCURSOS 10

typedef char str30[30] ;

typedef struct tRegEst {
    long int DNI;
    str30 carrera;
    int aniIngreso;
    str30 apellido ;
    str30 nombre ;
};
```

```
typedef tRegEst tVectorEst[_CANTEST] ;

typedef tVectorEst tVectorCur[_CANTCURSOS] ;
```

O la siguiente alternativa

```
typedef tRegEst tMatrizCur[_CANTCURSOS] [_CANTEST] ;
```

**ASÍ LAS COSAS,**

al primer índice lo  
llamamos fila y al  
segundo columna

```
int main(){
    tVectorCur vCursos;

    tMatrizCur mCursos;

    vCursos[0][0].DNI = 12;
    mCursos[0][0].DNI = 12;
}
```

**Entonces, independientemente de cómo se defina, la forma de acceso es la misma**

## Veamos el código en C++ con un problema

Se trata de procesar los 10.000 tiros de una ruleta por cada día, durante los 30 días de un mes.

Para esto se define una matriz de 30 filas que representan los 30 días y 37 columnas que representan los 37 números posibles (0 al 36). En cada celda se "cuenta" la cantidad de veces que se repite el número en ese día.

Para probar el algoritmo se utiliza la función rand (genera un número al azar) para generar cada uno de los "tiros".

En el proceso jugar se repite día por día (30 días) 10.000 tiros.

```
void jugar(tMatrizRul r, int cantDias, int cantNumeros, int cantBolas) {  
    //genera 1000 bolas diarias random, y acumula cuántas  
    //salieron de cada número  
    int nro;  
    srand (time(NULL));  
    for (int dia=0;dia<cantDias; dia++) {  
        for (int tiros=0;tiros<cantBolas;tiros++) {  
            nro = rand () % cantNumeros;  
            r[dia][nro]++;  
        }  
    }  
}
```

Toma la hora del procesador para generar una "semilla" y que el valor sea al azar

Genera números al azar entre 0 y 36 (cantNumeros recibe el valor 36)

Cuenta (suma 1) en la matriz, en la fila correspondiente al día y la columna correspondiente al número generado

De esta manera se genera una matriz que simula una ruleta con valores "al azar". Es lo más parecido a lo que sucede en la realidad. La cantidad de valores a generar (cantBolas) se pasa en la lista de parámetros.

```
#include <iostream>
#include <stdlib.h>
#include <time.h>
using namespace std;

#define _CANT36 37
#define _CANTDIAS 30
#define _CANTBOLAS 10000

typedef int tMatrizRul[_CANTDIAS][_CANT36] ;

int main(){
    setlocale(LC_CTYPE, "Spanish");
    int dia;
    int nro;
    tMatrizRul mRuleta;
    inicializarRuleta(mRuleta, _CANTDIAS, _CANT36) ;
    jugar(mRuleta, _CANTDIAS, _CANT36, _CANTBOLAS) ;
    mostrarRuletaMes(mRuleta, _CANTDIAS, _CANT36) ;
    dia = 3 ;
    mostrarRuletaDia(mRuleta, _CANTDIAS, _CANT36 , dia) ;
    nro = 8 ;
    mostrarRuletaNro(mRuleta, _CANTDIAS, _CANT36 , nro);
    return 0;
}
```

El código que presentamos genera la matriz que simula la ruleta, muestra los tiros realizados en los 30 días, muestra la cantidad de repeticiones de los 37 números el día 3 y cuántas veces se repitió el número 8 durante los 30 días.

Las pruebas referidas al día y el número son al solo efecto de ver cómo funciona el programa.

Les proponemos que modifiquen el programa para que muestre diferentes días y diferentes números.

**Recordá:** se debe asignar el valor 0 a cada elemento de la matriz

Procesa el juego, para la cantidad de días, la cantidad de números y la cantidad de bolas

Prepara y muestra para el día 3

Prepara y muestra para el número 8

Para inicializar una matriz se debe recorrer por fila, y dentro de cada fila por columna

```
void inicializarRuleta(tMatrizRul r, int cantDias,
                      int cantNumeros) {
    for (int dia=0; dia<cantDias; dia++)
        for (int nro=0; nro<cantNumeros; nro++)
            r[dia][nro] = 0 ;
}
```

Para recorrer una matriz se debe recorrer por fila y dentro de cada fila por columna

```
void mostrarRuletaDia(tMatrizRul r, int cantDias,
                     int cantNumeros, int dia) {
    printf("Los números del día %d\n", dia) ;
    for (int nro=0; nro<cantNumeros; nro++)
        printf("%3d |", nro);
    printf("\n");
    for (int nro=0; nro<cantNumeros; nro++)
        printf("%3d |", r[dia-1][nro]);
    printf("\n");
}
```

En rojo están resaltadas las líneas de código que permiten mostrar en la primera fila/columna los números de la ruleta y los días .

Para recorrer una matriz por una fila, se fija la misma y se recorre toda la columna

```
void mostrarRuletaNro(tMatrizRul r, int cantDias, int cantNumeros, int nro) {
    printf("El número %d en los 30 días \n", nro) ;
    for (int dia=0; dia<cantDias; dia++)
        printf("%3d |", dia+1);
    printf("\n");
    for (int dia=0; dia<cantDias; dia++)
        printf("%3d |", r[dia][nro]);
    printf("\n");
}
```

Para recorrer una matriz por una columna, se fija la misma y se recorre toda la fila

```
void mostrarRuletaMes(tMatrizRul r, int cantDias, int cantNumeros) {
    printf(" ");
    for (int nro=0; nro<cantNumeros; nro++)
        printf("%3d |", nro) ;
    printf("\n");
    for (int dia=0; dia<cantDias ; dia++) {
        printf("%3d |", dia+1) ;
        for (int nro=0; nro<cantNumeros; nro++)
            printf("%3d |", r[dia][nro]);
        printf("\n");
    }
}
```



En la imagen se observan todas las repeticiones de todos los números durante un mes y las repeticiones de los números del día 3 y las veces que salió el número 8 en todo el mes (día por día)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	281	286	266	265	256	302	276	268	283	280	296	262	262	258	250	261	248	262	268	258	299	264	263	266	258	266	291	275	275	293	273	262	281	255	288	261	242
2	272	281	275	279	276	253	283	268	287	262	256	271	272	285	248	270	282	291	265	242	263	301	287	261	282	277	289	266	254	256	255	246	288	267	274	276	240
3	263	270	272	250	251	271	272	289	261	281	282	254	269	265	242	263	259	262	291	256	262	269	284	269	273	260	277	278	294	264	283	305	275	283	256	272	273
4	246	268	291	270	274	240	274	282	236	251	271	279	306	266	240	281	283	259	276	258	268	275	272	307	259	260	291	251	286	254	270	292	253	263	258	292	298
5	255	251	284	280	294	276	299	249	267	250	249	243	271	273	260	263	256	261	286	285	276	274	276	260	280	245	265	286	252	270	293	286	258	315	251	259	302
6	254	255	290	288	251	269	264	252	258	296	287	292	287	296	247	246	261	281	276	271	277	265	257	273	269	266	276	272	246	243	276	253	267	288	262	287	302
7	302	279	271	269	258	249	281	262	287	279	297	238	258	262	296	265	257	261	267	278	273	258	271	286	279	240	277	277	265	262	262	299	286	246	270	254	279
8	298	259	257	246	256	262	261	245	273	243	247	266	285	281	260	285	267	253	273	268	268	294	262	284	269	264	277	313	309	279	272	257	277	287	284	252	267
9	287	297	257	269	265	288	256	231	279	272	259	287	287	259	266	255	277	293	290	261	271	251	283	283	235	246	256	261	308	296	269	272	264	275	289	262	244
10	254	287	292	294	252	286	274	281	282	237	279	283	245	295	270	298	229	271	291	258	299	294	259	281	272	242	262	262	243	280	254	274	267	293	267	255	238
11	273	278	254	236	276	288	278	267	267	246	256	274	275	278	265	262	278	291	254	253	234	287	257	274	285	276	259	277	302	279	309	262	262	279	256	281	272
12	289	272	258	255	267	259	289	250	289	282	264	251	282	237	303	257	268	275	263	299	254	283	265	270	275	271	285	268	284	276	259	263	260	259	281	283	255
13	259	263	267	263	279	278	272	314	260	300	281	251	298	254	215	295	279	274	288	291	289	272	254	270	281	265	287	275	241	261	270	271	209	247	297	258	272
14	257	286	276	272	277	274	249	262	254	301	277	251	253	269	245	292	289	290	292	275	241	288	254	250	266	261	274	262	276	292	275	271	258	286	294	263	248
15	256	256	260	245	269	271	286	274	273	265	282	260	299	298	281	267	285	279	298	263	235	254	250	275	254	247	285	278	248	297	294	289	250	268	264	257	288
16	274	276	273	282	267	278	282	254	241	285	260	250	295	274	262	297	265	278	263	295	272	290	266	259	259	267	258	257	260	264	257	275	261	280	299	255	270
17	269	251	219	261	276	285	259	268	288	277	282	281	272	261	280	257	255	279	273	283	289	247	272	255	283	273	254	251	262	252	263	278	266	289	295	299	296
18	269	267	291	264	232	253	272	285	282	273	249	300	273	268	244	266	305	285	281	275	252	279	250	273	272	283	279	251	272	269	267	261	267	286	281	267	257
19	271	251	269	249	278	247	292	291	250	281	279	299	242	273	272	278	281	273	251	262	295	270	267	256	280	279	270	279	275	271	282	261	275	266	250	265	270
20	258	258	266	278	260	261	249	278	241	271	265	262	255	267	276	266	262	293	273	283	275	248	288	246	250	279	282	301	255	268	299	278	282	278	291	255	303
21	276	274	287	313	235	294	293	282	261	253	269	271	296	241	269	296	256	287	275	265	253	265	278	284	268	272	280	262	256	239	258	266	253	264	284	264	261
22	278	297	273	243	252	274	282	277	262	272	298	285	278	266	272	272	281	262	282	298	265	243	261	275	273	254	260	255	253	279	272	276	282	250	235	281	282
23	274	291	275	286	272	274	243	255	266	267	270	263	267	255	268	266	240	252	310	280	244	286	261	267	302	288	283	269	261	298	269	251	287	282	276	258	244
24	273	262	272	264	305	275	267	287	274	277	273	261	276	214	277	250	307	279	285	246	268	276	296	235	275	270	287	267	271	256	261	265	304	249	277	247	272
25	252	283	266	298	256	293	256	240	246	285	259	256	260	286	276	268	251	292	292	249	273	242	253	258	261	290	276	296	270	285	283	270	259	279	263	306	272
26	258	247	283	278	242	292	272	264	266	284	243	251	300	289	262	277	271	265	280	281	269	264	264	294	242	271	280	288	264	286	289	271	270	257	281	247	258
27	293	282	273	274	257	264	319	261	263	251	259	277	282	254	294	258	276	253	278	285	280	280	264	278	258	267	266	262	268	247	276	262	274	298	266	243	258
28	250	275	280	258	248	255	273	253	268	280	276	273	263	279	273	277	254	275	290	273	276	299	274	286	236	257	280	265	275	246	274	267	289	311	262	248	282
29	247	290	267	298	280	266	254	261	270	266	284	273	242	288	293	285	291	269	267	261	280	241	304	265	258	230	275	286	274	267	283	296	241	278	245	242	283
30	268	271	277	248	268	287	279	249	270	282	287	254	284	256	258	295	313	255	283	269	264	251	279	256	268	260	286	254	280	279	277	255	255	272	256	282	273

Los números del día 3

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
263	270	272	250	251	271	272	289	261	281	282	254	269	265	242	263	259	262	291	256	262	269	284	269	273	260	277	278	294	264	283	305	275	283	256	272	273

El número 8 en los 30 días

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
283	287	261	236	267	258	287	273	279	282	267	289	260	254	273	241	288	282	250	241	261	262	266	274	246	266	263	268	270	270



Mostremos ahora la repeticiones de cada uno de los números (0 al 36) y mostremos también ordenadas por repeticiones

Es interesante conocer cuántas veces se repite cada número en la ruleta, al cabo de un mes si tiramos 10.000 bolas diarias.

Veamos los resultados y luego el código.

La primera lista nos muestra ordenado por número la cantidad de veces que salió cada número. Luego lo ordenamos por cantidad de veces y mostramos la lista en ese orden.

**¡¡UH!! Acabo de ver que el número CERO está más o menos en la mitad,  
o sea sale tantas veces como cualquier número.**

Hagamos otra cuenta: 10.000 bolas por día, por 30 días son 300.000 bolas en el mes.

Si la ruleta tiene 27 números, podemos dividir 300.000 por 37 y ver (si probabilísticamente todos salen en igual proporción) el resultado es CHAN, CHAN, 8.108

**QUIERE DECIR QUE EL NÚMERO CERO ESTÁ ENTRE LOS MÁS PROBABLES**

Muestra la cantidad de veces que salió cada número en el mes

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
8099	8062	7985	8141	8158	8035	8145	8063	8031	8062	8115	8192	8115	8229	8146	8103	8230	7963	8198	8089	8062	7968	8037	8161	8043	7976	8170	8252	8151	8111	8144	8180	8047	8120	8217	8032	8168

Muestra la cantidad de veces (ordenado) que salió cada número en el mes

17	21	25	2	8	35	5	22	24	32	20	1	9	7	19	0	15	29	10	12	33	3	30	6	14	28	4	23	36	26	31	11	18	34	13	16	27
7963	7968	7976	7985	8031	8032	8035	8037	8043	8047	8062	8062	8062	8063	8089	8099	8103	8111	8115	8115	8120	8141	8144	8145	8146	8151	8158	8161	8168	8170	8180	8192	8198	8217	8229	8230	8252

## Vamos al código C++

Estructura de datos para los números y sus repeticiones. Es importante "asociar" cada número con sus repeticiones, dado que cuando se lo ordene, pierde la asociación con el índice.

```
struct tJugadaNro {  
    int nro;  
    int veces;  
};  
  
typedef tJugadaNro tVectorRul[_CANT36];
```

Inicializa el vector  
con 0 en las  
"veces" y el  
número en "nro"

Recorre la matriz y  
asigna la cantidad de  
repeticiones de cada  
número en el vector

Muestra las  
repeticiones de cada  
número ordenado  
por número

Ordena por  
cantidad de  
repeticiones

Muestra las  
repeticiones de  
cada número  
ordenado por  
repeticiones

```
void mostrarRuletaNroOrd(tMatrizRul r, int  
                        cantDias, int cantNumeros) {  
    // totales por día ordenados de mayor a menor  
    tVectorRul vNros;  
    printf("Muestra la cantidad de veces que  
           salió cada número en el mes\n");  
    inicializaVectorNros(vNros, cantNumeros);  
    armaVector(r, vNros, cantDias, cantNumeros) ;  
    muestraVector(vNros, cantNumeros) ;  
    printf("Muestra la cantidad de veces  
           (ordenado) que salió cada número en el  
           mes\n");  
    ordenar(vNros, cantNumeros) ;  
    muestraVector(vNros, cantNumeros) ;  
}
```

## Vamos las funciones en C++

Se inicializa el vector con los números y cantidad de repeticiones. Para armarlo se utiliza el índice (el 0 con el 0, el 1 con el 1 y así sucesivamente. Luego al ordenar por el campo "veces" se debe "arrastrar" al número que corresponde, por lo tanto se lo inicializa con el mismo valor del índice.

```
void inicializaVectorNros(tVectorRul vNros,
                          int cantNumeros){
    for (int i=0;i<cantNumeros;i++) {
        vNros[i].nro = i;
        vNros[i].veces = 0;
    }
}
```

**Ordenar es una adaptación de los vistos clases atrás.**

```
void ordenar(tVectorRul vector, int cantVector) {
    tJugadaNro aux;
    int minimo;
    for(int i=0;i<cantVector;i++) {
        minimo = i ;
        for(int j=i+1;j<cantVector;j++)
            if(vector[j].veces<vector[minimo].veces)
                minimo = j ;
        aux=vector[i];
        vector[i]=vector[minimo];
        vector[minimo]=aux;
    };
}
```

Se "arma" el vector sumando la cantidad de repeticiones de cada número. Para esto se recorre la matriz por "nro" y dentro de cada número por "día" y se suma en la posición del vector correspondiente al número (coincide con el índice)

```
void armaVector(tMatrizRul r, tVectorRul vNros, int cantDias,
                int cantNumeros) {
    for (int nro=0; nro<cantNumeros; nro++)
        for (int dia=0; dia<cantDias ; dia++)
            vNros[nro].veces= vNros[nro].veces + r[dia][nro];
}
```

**Muestra el contenido del vector, toma el número del campo "nro" del mismo. Es importante esto último dado que cuando se lo ordena por el campo "veces", el índice no coincide con el número de la ruleta.**

```
void muestraVector(tVectorRul vNros, int cantNumeros) {
    for (int nro=0; nro<cantNumeros; nro++)
        printf("%4d | ", vNros[nro].nro);
    printf("\n");
    for (int nro=0; nro<cantNumeros; nro++)
        printf("%4d | ", vNros[nro].veces);
    printf("\n");
}
```

Para entretenernos veamos los resultados si ejecutamos el programa 12 veces (equivalente a un año)

```
Mes 1
10| 31| 0| 14| 1| 30| 27| 28| 8| 22| 18| 4| 5| 21| 20| 36| 35| 2| 19| 24| 16| 9| 7| 12| 3| 32| 17| 23| 15| 6| 29| 13| 11| 26| 25| 33| 34|
7902|7947|7978|7999|7999|8001|8006|8006|8009|8012|8026|8033|8035|8036|8049|8055|8073|8087|8088|8102|8110|8119|8139|8141|8162|8192|8197|8199|8199|8202|8209|8224|8237|8254|8307|8330|8336|
jugar
Mes 2
11| 29| 0| 1| 8| 16| 7| 35| 31| 26| 30| 27| 2| 3| 5| 10| 21| 34| 36| 15| 9| 33| 18| 12| 14| 32| 23| 20| 4| 17| 28| 24| 22| 13| 6| 25| 19|
7970|7971|8012|8012|8036|8036|8039|8060|8061|8062|8062|8064|8071|8082|8083|8085|8088|8096|8097|8107|8108|8112|8115|8118|8121|8132|8150|8154|8158|8159|8165|8166|8199|8224|8234|8238|8353|
jugar
Mes 3
4| 3| 29| 25| 18| 5| 12| 8| 32| 17| 14| 0| 21| 6| 11| 35| 34| 30| 22| 23| 16| 13| 33| 9| 36| 27| 7| 1| 28| 10| 26| 31| 15| 2| 20| 19| 24|
7906|7974|7978|7979|8007|8009|8011|8019|8027|8048|8053|8056|8058|8065|8083|8083|8084|8110|8124|8127|8129|8133|8135|8141|8146|8150|8156|8158|8171|8171|8190|8225|8227|8239|8263|8265|8300|
jugar
Mes 4
26| 18| 2| 36| 14| 21| 8| 6| 0| 32| 10| 5| 13| 30| 15| 23| 27| 31| 29| 33| 4| 28| 3| 35| 25| 7| 22| 17| 19| 16| 1| 34| 11| 20| 24| 9| 12|
7923|7954|7975|7977|8008|8009|8017|8040|8041|8041|8042|8045|8055|8069|8083|8085|8089|8098|8102|8114|8119|8136|8152|8153|8157|8164|8183|8185|8192|8208|8213|8214|8218|8218|8228|8243|8250|
jugar
Mes 5
1| 7| 10| 22| 13| 20| 26| 35| 5| 25| 14| 0| 28| 29| 12| 8| 24| 34| 3| 23| 6| 33| 15| 4| 32| 31| 2| 16| 18| 19| 30| 36| 27| 17| 21| 9| 11|
7857|7933|7998|8013|8020|8024|8031|8052|8054|8064|8072|8074|8076|8076|8080|8085|8090|8090|8115|8123|8124|8128|8129|8132|8137|8138|8163|8168|8173|8177|8182|8191|8207|8237|8256|8257|8274|
jugar
Mes 6
33| 28| 31| 26| 15| 34| 32| 3| 36| 5| 12| 16| 7| 13| 23| 35| 4| 22| 30| 17| 24| 2| 9| 11| 8| 14| 18| 27| 10| 25| 20| 19| 21| 6| 1| 0| 29|
7941|7962|7968|7969|7991|7997|8000|8012|8016|8038|8068|8081|8091|8099|8101|8103|8104|8109|8114|8117|8133|8135|8137|8140|8152|8154|8156|8161|8161|8172|8184|8211|8219|8231|8251|8256|8266|
jugar
Mes 7
30| 28| 13| 11| 10| 33| 36| 17| 8| 31| 2| 19| 18| 12| 7| 14| 24| 26| 6| 21| 16| 20| 29| 35| 3| 25| 9| 27| 23| 5| 15| 4| 22| 1| 32| 0| 34|
7968|7974|7992|7999|8007|8009|8010|8013|8025|8045|8048|8066|8069|8076|8080|8089|8101|8101|8106|8109|8114|8116|8119|8123|8135|8143|8154|8159|8166|8201|8214|8214|8217|8220|8244|8253|8321|
jugar
Mes 8
14| 30| 22| 5| 11| 24| 27| 2| 21| 18| 25| 28| 29| 32| 9| 15| 0| 19| 10| 1| 36| 6| 8| 7| 33| 26| 34| 17| 13| 12| 4| 31| 16| 35| 23| 20| 3|
7883|7945|7946|7970|7991|8003|8003|8012|8037|8044|8055|8065|8073|8092|8101|8102|8110|8112|8118|8129|8138|8146|8149|8156|8156|8173|8177|8178|8184|8189|8190|8191|8191|8212|8245|8248|8286|
jugar
Mes 9
29| 9| 26| 36| 27| 13| 14| 3| 16| 0| 22| 33| 17| 4| 23| 11| 31| 34| 35| 8| 32| 28| 5| 21| 15| 1| 2| 18| 20| 25| 24| 12| 19| 7| 10| 6| 30|
7956|7963|7965|7977|7979|7992|8000|8022|8048|8048|8052|8068|8070|8083|8090|8097|8107|8112|8119|8121|8131|8133|8134|8142|8142|8151|8156|8157|8162|8176|8213|8217|8222|8230|8236|8243|8286|
jugar
Mes 10
11| 29| 13| 31| 32| 19| 30| 14| 26| 12| 8| 33| 4| 23| 2| 5| 22| 21| 35| 34| 1| 28| 24| 7| 15| 10| 6| 16| 27| 36| 25| 17| 3| 0| 9| 18| 20|
7911|7975|7976|7989|7989|8013|8023|8024|8025|8031|8035|8061|8062|8072|8079|8100|8104|8106|8107|8116|8117|8124|8139|8140|8149|8153|8159|8163|8166|8176|8203|8204|8226|8229|8231|8307|8316|
jugar
Mes 11
34| 24| 35| 10| 28| 31| 36| 2| 15| 26| 6| 14| 13| 25| 29| 17| 22| 27| 9| 5| 3| 21| 1| 23| 12| 16| 11| 8| 20| 33| 0| 30| 4| 7| 32| 18| 19|
7920|7923|7983|7984|8006|8020|8020|8024|8026|8031|8032|8033|8051|8066|8071|8083|8092|8094|8096|8110|8112|8121|8122|8142|8166|8168|8182|8192|8198|8208|8211|8216|8232|8256|8263|8273|8273|
jugar
Mes 12
36| 22| 35| 33| 14| 34| 19| 28| 32| 31| 29| 12| 4| 18| 5| 27| 30| 11| 7| 6| 24| 1| 17| 9| 16| 21| 23| 0| 2| 26| 13| 25| 8| 15| 3| 10| 20|
7949|7985|7987|7993|8006|8016|8035|8035|8036|8041|8048|8053|8082|8082|8098|8102|8103|8105|8106|8108|8114|8129|8130|8131|8148|8149|8160|8163|8170|8171|8174|8183|8213|8222|8226|8241|8306|
```

Direccionamiento indirecto



## Aplicaciones con matrices y vectores: acceso indirecto

Supongamos ahora que se dispone de un vector con datos de 30 artículos que vende un importador de productos de telefonía. Los datos de cada producto que contiene el vector son: **código (4 caracteres alfanuméricos)**, **descripción (60 caracteres)**, **stock (número entero)**, **origen (país)**.

Además se conocen las ventas, cuyos datos son los siguientes : **fecha de venta (AAAAMMDD)**, **código del artículo** y **cantidad**.

**Se requiere hacer un reporte en forma de planilla, donde las filas representen los artículos y las columnas los 12 meses de año, que contenga la cantidad vendida de cada artículo en cada mes.**

**SI PRESTAN ATENCIÓN, SE PARECE (SOLO SE PARECE) A LA PLANILLA CON LOS NÚMEROS DE LA RULETA EN CADA MES.**

**Para ser más claro, se debe definir una matriz de 30 filas (una por cada artículo) y 12 columnas (una por cada mes) y al procesar cada venta, se debe "acumular" en la celda correspondiente al producto y al mes.**

**Pero,..., parece fácil.....¿lo es?, NOS DEBEMOS PREGUNTAR, ¿cómo se calcula la fila (0 a 29) a partir del código de artículo?, ¿y con la columna cómo hacemos?**

**Se trata de dos preguntas con dos respuestas diferentes:**

**Comencemos con el mes que parece más fácil: tenemos la fecha, entonces tenemos el mes, por lo tanto tomamos el mes, le restamos 1 y tenemos la columna. ¡¡BINGO!!**

**¿Y la fila?, también es fácil: tenemos un vector, tenemos el código, tenemos la función BUSCAR, entonces buscamos y el valor que devuelve es el índice buscado. ¡¡BINGO!!**

## Veamos el código en C++

### Definición de la estructura de datos, struct para artículo, fecha y datos a ingresar

```
// MODIFICAR LAS SIGUIENTES DEFINICIONES PARA AMPLIAR
ARTICULOS Y MESES
#define _CANTART 3
#define _CANTMESES 2

typedef char str60[61];
typedef char str4[5];

struct tRArt {
    str4 cod;
    str60 desc;
    int stok;
};

struct tRFecha {
    int d;
    int m;
    int a;
};

struct tRVenta {
    str4 cod;
    tRFecha fecha;
    int cant;
};

typedef tRArt tVectorArt[_CANTART];
typedef int tMatrizVentas[_CANTART][_CANTMESES];
```

En el main (recordar que debe haber un único main) se definen las variables que se utilizan en todo el programa. En este caso se deben definir el vector de artículos y la matriz donde se irán acumulando las ventas.

```
int main(){
    setlocale(LC_CTYPE, "Spanish");

    tMatrizVentas mVentas;
    tVectorArt vArticulos;

    inicializarVentas(mVentas, _CANTART, _CANTMESES);
    cargarArticulos(vArticulos, _CANTART);
    mostrarArt(vArticulos, _CANTART);
    procesarVentas(mVentas, _CANTART, _CANTMESES, vArticulos, _CANTART);
    mostrarReporte(mVentas, _CANTART, _CANTMESES, vArticulos, _CANTART);
    return 0;
}
```

Se definen las variables del tipo arreglos que se pasan a todas las funciones.

Se inicializan los arreglos,, se piden los artículos, se procesan las ventas, se muestran los resultados

Se define un vector para los artículos que contenga el código, la descripción y el stock.  
Se define una matriz que represente en las filas a los artículos y en las columnas a las filas y cuyo contenido es el acumulado de ventas del artículo en cada mes

**Se inicializa la matriz de ventas, se recorren las filas y por cada fila se recorre toda la columna**

```
void inicializarVentas(tMatrizVentas mVtas, int
cantArt, int cantDias) {
    for (int art=0; art<cantArt; art++)
        for (int dia=0; dia<cantDias; dia++)
            mVtas[art][dia] = 0 ;
}
```

**Se cargan en memoria los N artículos a procesar**

```
void cargarArticulos(tVectorArt vArt, int cantArt){
    for (int i=0;i<cantArt;i++) {
        printf("Ingresar código artículo (%3d de %3d), descripción y stock
de cada artículo separados por la tecla ENTER\n", i+1, cantArt);
        cin >> vArt[i].cod;
        cin >> vArt[i].desc;
        cin >> vArt[i].stok;
    }
}
```

**Separamos el mes de la fecha. Es importante realizarlo mediante una función, así si el proceso de "transformar" el código en un número es más complejo, se "toca" solo esta función.**

```
int separoMes(tRFecha f) {
    // retorna el mes - 1 para transformarlo en un índice
    int mes = f.m-1 return mes;
}
```

**El ya conocido por todos nosotros BUSCARVECTOR.**

```
int buscarVArt(tVectorArt vArt, int cantArtV, str4 cod){
    int i=0;
    while ((i<cantArtV)&&(strcmp(cod, vArt[i].cod)!=0))
        i++;
    if (i>=cantArtV) i = -1;
    return i;
}
```

**El también conocido RECORRERVECTOR.**

```
void mostrarArt(tVectorArt vArt, int cantArtV){
    printf("\nLista de artículos\n" );
    for (int i=0; i<cantArtV; i++)
        printf("%4s %60-s\n", vArt[i].cod, vArt[i].desc) ;
}
```

**Recorrido de la matriz, se muestra en un formato TABLA**

```
void mostrarReporte(tMatrizVentas mVtas, int cantArtM, int cantDias,
tVectorArt vArt, int cantArtV) {
    printf("    ");
    for (int dia=0; dia<cantDias; dia++)
        printf("%3d|", dia+1) ;
    printf("\n");
    for (int art=0; art<cantArtM ; art++) {
        printf("%4s |",vArt[art].cod) ;
        for (int dia=0; dia<cantDias; dia++)
            printf("%3d|", mVtas[art][dia]);
        printf("\n");
    }
}
```

# Ejemplo con 3 artículos y 2 meses

Prueba de escritorio con los siguientes datos:

Artículos  
L001 Resma 20  
L002 Cuaderno 30  
L003 Birome 25

Ventas  
L001 2/1/2020 10  
L001 2/2/2020 5  
L002 3/1/2020 12

Los resultados esperados son:

L001 Resma 10 en enero, 2 en febrero  
L002 Cuaderno 12 en enero

Les proponemos ampliar la prueba para más artículos y meses.

```
Lista de artículos
L001
L002
L003
Resma
Cuaderno
Birome
Ingresar las ventas, con código = 0 finalizan
Ingresar código de artículo (0=fin)
L001
2
1
2020
Ingresar cantidad
10
pedirVta:ingresó L001 2/ 1/2020 10
Ingresar código de artículo (0=fin)
L001
2
2
2020
Ingresar cantidad
5
pedirVta:ingresó L001 2/ 2/2020 5
Ingresar código de artículo (0=fin)
L002
3
1
2020
Ingresar cantidad
12
pedirVta:ingresó L002 3/ 1/2020 12
Ingresar código de artículo (0=fin)
0
pedirVta:ingresó 0 3/ 1/2020 12
1| 2|
L001 | 10| 5|
L002 | 12| 0|
L003 | 0| 0|
```