



PROGRAMACION I APUNTES: ARREGLOS

«Fíate del Señor de todo tu corazón,
y no te apoyes en tu propia prudencia.
Reconócelo en todos tus caminos,
y Él enderezará tus veredas.»
Proverbios 3:5-6

Profesora: DELY M. GIL A.
VALENCIA, Febrero, 2010

ARREGLOS

Un arreglo es una estructura de datos en la que se almacena una colección de datos del mismo tipo (por ejemplo las notas de los alumnos). Al tipo se le llama *tipo base* del arreglo. Los datos individuales se llaman *elementos del arreglo*.

TIPOS DE ARREGLOS

Los arreglos pueden ser:

- Unidimensionales, también llamados Vectores o listas
- Bidimensionales, denominados tablas o matrices.
- Multidimensionales, con tres o más dimensiones.

Dely Gil

CARACTERÍSTICAS DE LOS ARREGLOS

Un arreglo se caracteriza por :

1. Almacenar los elementos del arreglo en posiciones de memoria continua
2. Tener un único nombre de variable que representa a todos los elementos (Notas), y éstos a su vez se diferencian por un índice o subíndice.
Notas[0], ..., Notas[n-1] {Lenguaje C}
Notas[1]..Notas[n] {Pascal}
3. Acceso directo o aleatorio a los elementos individuales del arreglo. Notas[5]= 15;

Dely Gil

ARREGLO UNIDIMENSIONAL

Un arreglo de una dimensión (vector o lista) es un tipo de datos estructurado compuesto de un número de elementos finitos, consecutivos de tamaño fijo y elementos homogéneos.

Finito indica que el arreglo tiene un número determinado de elementos. *Consecutivo* que se almacena en posiciones consecutivas de memoria, *Tamaño fijo* significa que el tamaño del arreglo debe ser conocido en tiempo de compilación, *homogéneo* significa que todos los elementos son del mismo tipo.

Dely Gil

INICIALIZACIÓN DE UN ARRAY

Cuando se inicializa un arreglo, el tamaño del arreglo se puede determinar automáticamente por las constantes de inicialización

```
int edades[6] = { 12,14,11,16,11,9};
```

```
int A[] = { 1,2,4}; //arreglo de 3 elementos
```

```
char c[] = { 'J', 'o', 's', 'u', 'e'}; //arreglo de 5 elementos
```

```
char s[] = "La batalla de la FE"; //arreglo de caracteres
// almacena en la última posición el carácter NULO
```

Dely Gil

FORMATO DE DECLARACIÓN DE UN ARREGLO UNIDIMENSIONAL

tipo nombre_array[tipo subíndice];

```
#define TAM 10
```

```
#define MAXCAR 80
```

```
#define NHORAS 24
```

```
#define NFIL 24
```

```
int main(){
```

```
    int arreglo1[] = {1,2,3,4,5};
```

```
    int arreglo2[5] = {1,2,3,4,5};
```

```
    int arreglo3[8] = {1,2,3,4,5};
```

```
    int x[TAM];
```

```
    char texto[MAXCAR];
```

```
    double temperaturas[NHORAS];
```

```
    int asientos[NFIL];
```

```
}
```

Dely Gil

OPERACIONES CON ARREGLOS

1. Lectura de un vector

```
for( int i=0; i<=N;i++){
    scanf("%d",&Notas[i]);
```

2.- Escritura de un Vector

```
for( int i=0; i<=N;i++){
    printf("Notas[%d] = %d ",i,Notas[i]);
```

4.- Búsqueda

- Lineal (Secuencial) {No ordenados}
- Binaria para arreglos ordenados

5.- Ordenamiento

- Inserción, Selección, Burbuja o Intercambio, Shell, QuickSort (Ordenación rápida), Ordenación por Fusión o Mezcla

6.- Inserción y eliminación

Dely Gil

ARREGLO BIDIMENSIONAL

Un arreglo bidimensional (tabla o matriz) es un arreglo con dos índices, al igual que los vectores que deben ser ordinales o tipo subrango.

		Columnas		
		0	1	2
Filas	0	A[0,0]	A[0,1]	A[0,2]
	1			
	2			
	3	A[0,3]	A[3,3]	A[3,2]

Para localizar o almacenar un valor en el arreglo se deben especificar dos posiciones (dos subíndices), uno para la fila y otro para la columna.

Dely Gil

MANIPULACIÓN DE TABLA

Recorrido por fila

```
for(i=0; i<N; i++){
    for(j=0; j<M; j++){
        printf("Ingrese el valor del elemento [%i][%i] :", i+1, j+1);
        scanf("%f", &matriz[i][j]);
    }
}
```

Recorrido por Columnas

```
for(j=0; j<M; j++){
    for(i=0; i<N; i++){
        printf("Ingrese el valor del elemento [%i][%i] :", i+1, j+1);
        scanf("%f", &matriz[i][j]);
    }
}
```

Dely Gil

FORMATO DE ARREGLO BIDIMENSIONALES

tipo identificador [índice1] [índice 2];

```
#define N 5
#define M 3
```

```
int main(){
    int matriz[N][M]:
    int matriz[4][2] = {
        { 11, 12},
        { 21, 22},
        { 31, 32},
        { 41, 42}
    }
}
```

Dely Gil

ARREGLOS PARALELOS

Dos o más arreglos que utilizan el mismo subíndice para referirse a términos homólogos se llaman arreglos paralelos.

Nombres[0]	Job	Edades[0]	82
Nombres[1]	Josue	Edades[1]	28
	.		.
	.		.
	.		.
Nombres[n-1]	Isaias	Edades[n-1]	9

Dely Gil

```
#include <stdlib.h>
#include <ctype.h>
#include <stdio.h>
#define TAM 10

int main(){
    char Min[TAM], May[TAM];
    int i;

    for(i=0;i<TAM;i++){
        Min[i] = getchar();
        May[i] = toupper(Min[i]);
    }

    puts("El vector en mayuscula");
    for(i=0;i<TAM;i++){
        putchar(May[i]);
        printf("\n");
    }
    system("PAUSE");
    return 0;
}
```

Dely Gil

```
#include<stdio.h>
#include<stdlib.h>
#include <ctype.h>
#define MAX 4

void LeerArray(char sList[]);
void Transf(char sList[],char sList2[]);
void Imprimir(char sList[]);
int i;

int main(){
    char Min[MAX],May[MAX];

    LeerArray(Min);
    Transf(Min,May);
    Imprimir(May);
    system("PAUSE");
}

void LeerArray(char sList[]){
    for ( i=0; i<MAX; i++)
        sList[i] = getchar();
}

void Transf(char sList[],char sList2[]){
    for(i=0;i<MAX;i++)
        sList2[i] = toupper(sList[i]);
}

void Imprimir(char sList[]){
    for(i=0;i<MAX;i++){
        putchar(sList[i]);
        printf("\n");
    }
}
```

Dely Gil

ARREGLOS COMO PARÁMETROS FORMALES

- Se pasan como parámetros por referencia (dirección); por lo tanto sus elementos pueden cambiar.

```
#define MAX 10
void modificar(float datos[MAX])
```

- Se pasa sólo la dirección de la celda de memoria donde comienza el arreglo. Este valor se representa por el nombre del arreglo.
- El tamaño puede o no especificarse en la declaración.

```
float media(float datos[], int n)
```
- Si no se especifica el tamaño del arreglo, se puede pasar otro argumento que especifique el tamaño.
- Si no se escriben los nombres de los argumentos entonces una pareja vacía de corchetes debe seguir el tipo de dato del argumento array.

```
float media(float [], int )
```
- En arreglos bidimensionales no es necesario indicar la dimensión de las filas pero sí la dimensión de las columnas.

```
float Suma(float datos[][MAX], int )
```

Dely Gil

ARREGLOS COMO PARÁMETROS ACTUALES

- Se indica solo con el nombre o identificador sin corchetes ni índices.

```
#include <stdio.h>
#define MAX 10
//prototipos
void modificar(float datos[MAX]);
float media(float datos[], int n)
```

```
int main(){
    float a[MAX], resMedia;
    modificar(a);
    resMedia = media(a,MAX);
}
```

```
void modificar(float datos[MAX]) {
```

```
//Complete el código
```

```
}
```

```
float media(float datos[], int n) {
```

```
//Complete el código
```

```
}
```

Dely Gil

Ingresar valores en un vector v. Almacenar en un vector A todos los números negativos y en un vector B todos los positivos o iguales a cero

```
#include <stdlib.h>
#include <ctype.h>
#include <stdio.h>
#define MAX 5

int main(){
    int v[MAX],a[MAX],b[MAX],j=0,k=0,i=0;

    while(i<MAX) {
        printf("Ingrese el valor del elemento %i :",i+1);
        scanf("%i",&v[i]);
        if (v[i]<0) {
            a[j]=v[i] ;
            j++;
        }else{
            b[k]=v[i] ;
            k++;
        }
        i++;
    }
    for(i=0;i<j;i++)
        printf("%i\n",a[i]);
    puts("El vector positivo");
    for(i=0;i<k;i++)
        printf("%i \n",b[i]);
    system ("PAUSE");
    return 0;
}
```

Dely Gil

```
//Cabeceras
void LeerArray(int Y[]);
void generar(int Y[],int X[],int Z[],int &tX,int &tZ);
void Imprimir(int W[],int tam);

int main(){
    int v[MAX],a[MAX],b[MAX],j=0,k=0,i=0;
    LeerArray(v);
    generar(v,a,b,j,k);
    puts("Valores Positivos");
    Imprimir(a,j);
    puts("Valores Negativos");
    Imprimir(b,k);
    system ("PAUSE");
    return 0;
}

void LeerArray(int A[]){
    int i=0;
    while(i<MAX) {
        printf("Ingrese el valor del elemento %i :",i+1);
        scanf("%d",&A[i]);
        i++;
    }
}

void generar(int Y[],int X[],int Z[],int &tX,int &tZ){
    int i=0;
    while(i<MAX) {
        if (Y[i]<0) {
            X[tX]=Y[i] ;
            tX++;
        }else{
            Z[tZ]=Y[i] ;
            tZ++;
        }
        i++;
    }
}
```

Dely Gil

Genere un vector de tamaño 10 con números reales leídos desde teclado. Calcule el promedio e indique cuantos elementos del vector son mayores que el promedio y cuantos menores o iguales.

```
#include <stdlib.h>
#include <ctype.h>
#include <stdio.h>
#define MAX 5

int main(){
    float v[MAX], suma=0, promedio;
    int i, contmay=0, contmen=0;

    for ( i=0; i<MAX; i++){
        printf("Ingrese el valor del elemento %i :", i+1);
        scanf("%f", &v[i]);
        suma += v[i];
    }
    promedio = suma/MAX;
    // Contar mayores y menores
    i=0;
    while (i<MAX){
        if (v[i] > promedio)
            contmay++;
        else
            contmen++;
        i++;
    }
    printf("La cantidad de elementos mayores que el promedio son %d", contmay);
    printf("La cantidad de elementos menores o iguales al promedio son %d", contmen);
    system ("PAUSE");
    return 0;
}
```

Dely Gil

```
#include <stdlib.h>
#include <ctype.h>
#include <stdio.h>
#define MAX 5

void LeerArray(float a[]);
float Suma(float a[]);
float Promedio(float sum);
void MayMen(float a[], float prom, int &may, int &men);
void Salida(float a[], float sum, float p, int may, int men);

int i;
int main(){

    float v[MAX], resultSum=0, resultPro;
    int i, contmay=0, contmen=0;

    LeerArray(v);
    resultSum = Suma(v);
    resultPro = Promedio(resultSum);
    MayMen(v, resultPro, contmay, contmen);
    Salida(v, resultSum, resultPro, contmay, contmen);
    system("PAUSE");
    return 0;
}

void LeerArray(float a[]){
    for ( i=0; i<MAX; i++){
        printf("Ingrese el valor del elemento %i :", i+1);
        scanf("%f", &a[i]);
    }
}
```

Dely Gil

```

float Suma(float a[]){
    float total=0;
    for ( i=0; i<MAX; i++)
        total += a[i];
    return total;
}

float Promedio(float sum){
    return sum/MAX;
}

void MayMen(float a[],float p,int &may, int &men){
    int i=0;

    while (i<MAX){
        if (a[i] > p)
            may++;
        else
            men++;
        i++;
    }
}

void Salida(float a[],float sum,float p, int may,int men){
    for ( i=0; i<MAX; i++)
        printf("El valor %i es %f\n:",i+1,a[i]);
    printf("La suma es %f: ", sum);
    printf("El promedio es %f: ", p);
    printf("La cantidad de elementos mayores que el promedio son %d",may);
    printf("La cantidad de elementos menores o iguales al promedio son %d",men);
}

```

Dely Gil

```

/*El siguiente programa captura 10 edades y nombres por medio de arreglos paralelos*/
#include <stdio.h>
#include <stdlib.h>
#include<strings.h>

#define PERS 3

int main(){
    int edades[PERS];
    char nombres[PERS] [25];
    char may_nom[25];
    int may=0,i;

    for(i=0;i<PERS;i++){
        printf("Persona %i\n",i+1);
        printf("Nombre: ",i+1);
        fflush( stdin );
        scanf("%s",nombres[i]);
        printf("\nEdad:");
        scanf("%i",&edades[i]);
    }
    for(i=0;i<PERS;i++){
        if (edades[i]>= may){
            may = edades[i];
            strcpy(may_nom,nombres[i]);
        }
    }
    printf("La persona %s tiene la mayor edad %i", may_nom, may);
    system("PAUSE");
    return 0;
}

```

//Implemente el ejercicio con Subprogramas

Dely Gil

/*El siguiente programa captura las calificaciones de 5 alumnos en 3 exámenes, y despliega en pantalla los promedios */

		Exam 1	Exam 2	Exam 3	Promedio
Alumnos[0]	Job	10	10	10	10
Alumnos[1]	Josue	20	20	20	20
Alumnos[2]	Luis	15	15	15	15
Alumnos[3]	Isaias	05	05	05	05

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

```
#define M 2
#define N 3
```

```
int main(){
    float Examenes[M][N];
    char Alumnos[M][25];
    float Promedios[M];

    float suma;
    int i,j;

    //lectura de arrays paralelos de manera simultánea
    puts("Nombre Examen1 Examen2 Examen3 Promedio");
    for(i=0;i<M;i++){
        printf("Alumno %i\n",i+1);
        fflush( stdin );
        printf("Nombre: ");
        scanf("%s",Alumnos[i]);
```

Dely Gil

```
        suma =0;
        for(j=0;j<N;j++){
            printf("\nExamen %i: ",j+1);
            scanf("%f",&Examenes[i][j]); //lectura de matriz
            suma += Examenes[i][j];
        }
        Promedios[i] =suma/N;
    }
```

```
    printf("El promedio es :");
    for(i=0;i<M;i++){
        printf("\nAlumno %i : %f",i+1,Promedios[i]);
        system("PAUSE");
    }
    return 0;
}
```

//Implemente el ejercicio con Subprogramas

Calcule lo siguiente:

- 1.- El promedio por cada asignatura.**
- 2.- La mayor nota en cada asignatura. Debe mostrar el nombre del alumno.**
- 3.- El alumno de mayor promedio.**

Dely Gil

```
//Promedio por asignatura
for(j=0;j<N;j++){
    suma =0;
    for(i=0;i<M;i++)
        suma += Examenes[i][j];
    Asignatura[j] = suma/M;
}

printf("\nEl promedio por Asignatura es :");
for(i=0;i<N;i++)
    printf("\nExamen %i : %f",i+1,Asignatura[i]);

//la mayor nota de cada asignatura
for(j=0;j<N;j++) {
    may = 0;
    for(i=0;i<M;i++)
        if (Examenes[i][j]>=may){
            may = Examenes[i][j];
            strcpy(may_nom,Alumnos[i]);
        }
    printf("\nLA nota mayor del examen %d es %f y el nombre
es %s",j+1,may,may_nom);
}

//el mayor promedio
may=0;
for(i=0;i<M;i++)
    if (Promedios[i]>may)
        may = Promedios[i];
printf("La mayor nota del promedio es: %f ",may);
system("PAUSE");
return 0;
```

Dely Gil

En los siguientes países: Colombia, Venezuela y España se presentan cuatro(4) grupos de música cristiana. Ingresar los nombres de los grupos (Ejemplo: Rojo, Rescate, Veinte Veinte, Hillsong,..., entre otros). El costo de los boletos de cada grupo son 10,20,30 respectivamente. Calcular:

1. Calcular el total de boletos por cada concierto.
2. Calcular el total de boletos por países.
3. Calcular el ingreso por cada concierto.
4. El país con mayor venta de boletos.
5. El porcentaje de venta por cada grupo cristiano.

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

```
#define C 3
#define P 3
```

```
int main(){
    int arreglo1[] = {1,2,3,4,5} ;
    int arreglo2[5] = {1,2,3,4,5} ;
    char Pais[3][25] = { "Colombia", "Vzla", "España"},
    may_pais[25];
    float Precios[C]={10,20,30};
    char Conciertos [C] [25];
    int Boletos[C][P],TotPais[P],TotConc[C],suma=0;
    float Costo[C];
    int i,j,may=0;
```

Dely Gil

```
//Llenar Matriz y Vector Conciertos: Arreglos paralelos
```

```
for(i=0;i<C;i++){
    printf("Ingrese nombre del Grupo :");
    fflush( stdin );
    scanf("%s",&Conciertos[i]);
    for(j=0;j<P;j++){
        printf("Cantidad de Boletos en el Pais %s: ",Paises[i]);
        scanf("%i",&Boletos[i][j]);
    }
}
```

```
//Imprimir los Bletos
```

```
printf("\nCantidad de Boletos en los Paises\n ");
```

```
for(i=0;i<C;i++){
    printf("\n");
    for(j=0;j<P;j++){
        printf("%i\t",Boletos[i][j]);
    }
}
```

```
//Cant total x Conciertos
```

```
for(i=0;i<C;i++){
    suma =0;
    for(j=0;j<P;j++){
        suma += Boletos[i][j];
    }
    TotConc[i]=suma;
}
```

```
printf("\nCantidad de Boletos pos Conciertos \n");
```

```
for(i=0;i<C;i++) {
    printf("%i\t",TotConc[i]);
}
```

Dely Gil

```
//Cant total x Paises
```

```
for(j=0;j<P;j++) {
    suma =0;
    for(i=0;i<C;i++){
        suma += Boletos[i][j];
    }
    TotPais[j]=suma;
}
```

```
//Imprimir cant de boletos de los paises
```

```
printf("\nCantidad de Boletos por Paises\n");
```

```
for(i=0;i<P;i++) {
    printf("%i\t",TotPais[i]);
}
```

```
//Ventas
```

```
for(i=0;i<C;i++){
    Costo[i] = TotConc[i]*Precios[i];
    printf("\nEl costo %i es : %f",i+1,Costo[i]);
}
```

```
//Imprimir cant de ingresos por conciertos
```

```
printf("\nCantidad de ingresos por conciertos:\n");
```

```
for(i=0;i<C;i++) {
    printf("%f\t",Costo[i]);
}
```

```
//Pais con mas ventas de Boletos
```

```
for(i=0;i<P;i++){
    if (TotPais[i]>=may){
        may = TotPais[i];
        strcpy(may_pais,Paises[i]);
    }
}
```

```
printf("\nEl pais %s tuvo la mayor venta es boletos con %i ",may_pais,may);
```

```
system("PAUSE");
```

```
return 0;
```

}

Dely Gil

MANEJO DE MATRICES

Diagonal Principal ($i = j$)

A00	A01	A02	A03	A04	A05
A10	A11	A12	A13	A14	A15
A20	A21	A22	A23	A24	A25
A30	A31	A32	A33	A34	A35
A40	A41	A42	A43	A44	A45
A50	A51	A52	A53	A54	A55

```
for(i=0;i<N;i++)
  for(j=0;j<N;j++)
    if (i==j)
      A[i][j] = 0;
```

```
for(i=0;i<N;i++)
  A[i][i] = 0;
```

Dely Gil

MANEJO DE MATRICES

Diagonal Secundaria ($i+j = N-1$)

A00	A01	A02	A03	A04	A05
A10	A11	A12	A13	A14	A15
A20	A21	A22	A23	A24	A25
A30	A31	A32	A33	A34	A35
A40	A41	A42	A43	A44	A45
A50	A51	A52	A53	A54	A55

```
for(i=0;i<N;i++) {
  J = N - i + 1;
  A[i][J] = 0;
}
```

```
for(i=0, j=N;i<N;i++, j--)
  if (i+j == N-1)
    A[i][j] = 0;
```

Dely Gil

MANEJO DE MATRICES

Diagonal Principal Superior ($i < j$)

A00	A01	A02	A03	A04	A05
A10	A11	A12	A13	A14	A15
A20	A21	A22	A23	A24	A25
A30	A31	A32	A33	A34	A35
A40	A41	A42	A43	A44	A45
A50	A51	A52	A53	A54	A55

```
for(i=0;i<N-1;i++)
  for(j=i+1;j<N;j++)
    A[i][j] = 0;
```

```
for(i=0;i<N;i++)
  for(j=0;j<N;j++)
    if (i<j)
      A[i][j] = 0;
```

Dely Gil

MANEJO DE MATRICES N=6

Diagonal Secundaria Superior ($i+j < N-1$)

A00	A01	A02	A03	A04	A05
A10	A11	A12	A13	A14	A15
A20	A21	A22	A23	A24	A25
A30	A31	A32	A33	A34	A35
A40	A41	A42	A43	A44	A45
A50	A51	A52	A53	A54	A55

```
for(i=0;i<N-2;i++)
  for(j=0;j<N-2-i;j++)
    A[i][j] = 0;
```

```
for(i=0;i<N;i++)
  for(j=0;j<N;j++)
    if (i+j < N-1)
      A[i][j] = 0;
```

Dely Gil

MANEJO DE MATRICESDiagonal Principal Inferior ($i > j$)

A00	A01	A02	A03	A04	A05
A10	A11	A12	A13	A14	A15
A20	A21	A22	A23	A24	A25
A30	A31	A32	A33	A34	A35
A40	A41	A42	A43	A44	A45
A50	A51	A52	A53	A54	A55

```
for(i=1;i<N;i++)
  for(j=0;j<i-1;j++)
    A[i][j] := 0;
```

```
for(i=0;i<N;i++)
  for(j=0;j<N;j++)
    if (i>j)
      A[i][j] = 0;
```

Dely Gil

MANEJO DE MATRICESDiagonal Secundaria Inferior ($i+j > N-1$)

A00	A01	A02	A03	A04	A05
A10	A11	A12	A13	A14	A15
A20	A21	A22	A23	A24	A25
A30	A31	A32	A33	A34	A35
A40	A41	A42	A43	A44	A45
A50	A51	A52	A53	A54	A55

```
for(i=1;i<N;i++)
  for(j=N-i;j<N;j++)
    A[i][j] = 0;
```

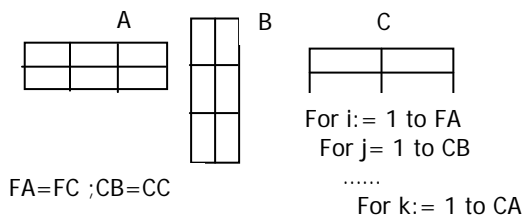
```
for(i=0;i<N;i++)
  for(j=0;j<N;j++)
    if (i+j > N-1)
      A[i][j] = 0;
```

Dely Gil

MANEJO DE MATRICES

Producto de Matrices

```
for(i=0;i<N;i++)
  for(j=0;j<N;j++){
    prod[i][j] = 0;
    for(k=0;k<N;k++)
      prod[i][j] = prod[i][j] + a[i][k]*b[k][j];
  }
```



Dely Gil

MATRIZ SIMÉTRICA

Una matriz A se dice que es simétrica si $A(i,j) = A(j,i)$ para todo i,j dentro de los límites de las matriz. (MXM)

MATRIZ ASIMÉTRICA

$$A(i,j) = -A(j,i)$$

MATRIZ TRASPUESTA

Una matriz M se dice que es traspuesta (MT) si $M(i,j) = MT(j,i)$ para todo i,j dentro de los límites de las matriz. (MXN)

Dely Gil

ORDENAMIENTO BURBUJA (BUBBLE)

Consiste en realizar pasadas sucesivas direccionando desde el primer elemento hasta el penúltimo, comparando cada uno de ellos con el siguiente. Esta versión del método va colocando en cada pasada el menor elemento de los tratados en la primera posición (burbujean), y el mayor al fondo del arreglo (hunden).

Dely Gil

Gráficamente podemos observar lo siguiente:

Vector

23	19	45	15
----	----	----	----

Pasada 1: $i:=1$

23	19	45	15
----	----	----	----

J=1

19	23	45	15
----	----	----	----

J=2

19	23	45	15
----	----	----	----

J=3

19	23	15	45
----	----	----	----

Dely Gil

Pasada 2: $i:=2$

19	23	15	45
----	----	----	----

J=1

19	23	15	45
----	----	----	----

J=2

19	15	23	45
----	----	----	----

Pasada 3: $i:=3$

19	15	23	45
----	----	----	----

J=1

15	19	23	45
----	----	----	----

Observar:

- Se usan 3 pasadas para arreglos de 4 elementos, es decir, $n-1$ pasadas.
- **LAS PASADAS SON $i:=N-1$**
- Para la pasada 1 ($i=1$) se realizan comparaciones ($j=1,2,3$)
- Para la pasada 2 ($i=2$) se realizan comparaciones ($j=1,2$)
- Para la pasada 3 ($i=3$) se realizan comparaciones ($j=1$)
- **LAS COMPARACIONES SON: $1..N-i$**

Dely Gil

ALGORITMO A

//ordenación de un vector que almacena edades

```

for(i=0;i<N-1;i++)
  for(j=i+1;j<N-i;j++)
    if (edades[j]>edades[j+1]) {
      aux      =edades[j];
      edades[j] =edades[j+1];
      edades[j+1] =aux;
    }

```

Dely Gil

ALGORITMO B

//ordenación del ejercicio de edades

```

for(i=0;i<N-1;i++)
  for(j=i+1;j<N;j++)
    if (edades[i]>edades[j]) {
      aux      =edades[i];
      edades[i] =edades[j];
      edades[j] =aux;
    }

```

Dely Gil

Gráficamente podemos observar lo siguiente:**Vector**

23	19	45	15
----	----	----	----

Pasada 1: i:= 1

23	19	45	15
----	----	----	----

J=2

19	23	45	15
----	----	----	----

J=3

19	23	45	15
----	----	----	----

J=4

15	23	45	19
----	----	----	----

Pasada 2: i:= 2

15	23	45	19
----	----	----	----

J=3

15	23	45	19
----	----	----	----

J=4

15	19	45	23
----	----	----	----

Pasada 3: i:= 3

19	19	45	23
----	----	----	----

J=4

Dely Gil

BÚSQUEDA LINEAL: Recorre el arreglo hasta encontrar el elemento a buscar. Si encuentra el elemento devuelve la posición y la variable lógica en verdadero (En Lenguaje C variable tipo int).

```
#define FALSE 0
```

```
#define TRUE 1
```

```
...
```

```
int Busqueda(int A[MAX],int elemento) {
```

```
    Encontrado = FALSE ;
```

```
    int i = 0;
```

```
    while ( (i<= MAX-1 ) && (!Encontrado) )
```

```
        (A[i]==elemento)? Encontrado = TRUE : i++;
```

```
    (Encontrado) ? return(i): return(-1);
```

```
}
```

Dely Gil

UNIÓN DE VECTORES

Pasos a seguir:

- ❖ Recorrer el primer arreglo (A)
- ❖ Buscar que no esté en el vector de Unión (C) para ser insertado en C.
- ❖ Recorrer el segundo arreglo (b)
- ❖ Buscar que no esté en el vector de Unión (C) para ser insertado en C.

10	20	30	40	50
----	----	----	----	----

A

2	10	4	10	50
---	----	---	----	----

B

10	20	30	40	50	2	4
----	----	----	----	----	---	---

C

Dely Gil

BÚSQUEDA BINARIA: Antes de ejecutar esta búsqueda es necesario que el vector esté ordenado.

```
primero=0 ; ultimo =n-1; encontrado= 0;
```

```
while ((primero<= ultimo ) && (!encontrado)) {
```

```
    central = (primero+ultimo)/2;
```

```
    if (elemento == A[central])
```

```
        encontrado = 1;
```

```
    else
```

```
        if (elemento > A[central])
```

```
            primero= central +1;
```

```
        else
```

```
            ultimo = central -1
```

```
}
```

```
(!encontrado) ? return (-1); return(central);
```

Dely Gil

INTERSECCIÓN DE VECTORES

Pasos a seguir:

- ❖ Recorrer el primer arreglo (A)
- ❖ Buscar el elemento en (B)
- ❖ Si el elemento está en B
 - Buscar que no esté en C para ser insertado.

10	20	30	40	50
----	----	----	----	----

A

2	10	4	10	50
---	----	---	----	----

B

10	50
----	----

C

Dely Gil

INSERCIÓN DE ELEMENTOS EN VECTORES

Requiere de cuatro acciones:

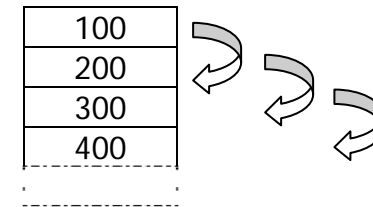
- Verificar si hay espacio en el vector.
- Desplazar los elementos para liberar el espacio para el valor a insertar
- Asignar el valor en la posición
- Modificar la dimensión del arreglo

Casos de Inserción:

- Al inicio del arreglo
- Entre elementos del arreglo
- Al final del arreglo

Dely Gil

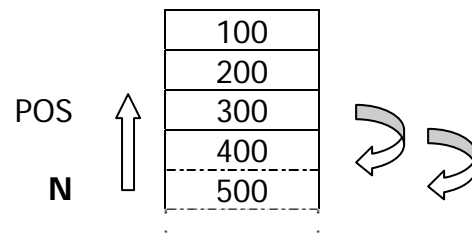
INSERCIÓN AL INICIO DEL ARREGLO (POS = 0)



```
for(int i=N;i>0;i--)
    Vector[i+1] = Vector[i];
Vector[1] := Valor
N++;
```

Dely Gil

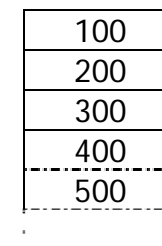
INSERCIÓN ENTRE ELEMENTOS (1 > POS < N)



```
for(int i=N;i>POS;i--)
    Vector[i+1] = Vector[i];
Vector[POS] = Valor
N++;
```

Dely Gil

INSERCIÓN AL FINAL DEL ARREGLO (POS = N)



```
Vector[POS] = Valor
N++;
```

Dely Gil

ELIMINACIÓN DE ELEMENTOS EN VECTORES

Requiere de dos acciones:

- Desplazar los elementos para eliminar el valor
- Modificar la dimensión del arreglo

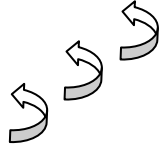
Casos de Eliminación:

- El primer elemento del arreglo (POS=1)
- Entre elementos del arreglo
- Al final del arreglo (POS = N)

Dely Gil

ELIMINACIÓN AL INICIO DEL ARREGLO (POS = 1)

100
200
300
400

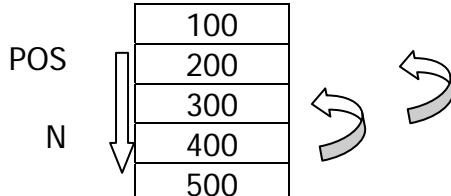


```
for(int i=2;i<=N-1;i++)
    Vector[i] = Vector[i+1];
N--;
```

Dely Gil

ELIMINACIÓN ENTRE ELEMENTOS (1 > POS < N)

100
200
300
400
500



```
for(int i= POS;i<=N-1;i++)
    Vector[i] = Vector[i+1];
N--;
```

Dely Gil

ELIMINACIÓN AL FINAL DEL ARREGLO (POS = N+1)

100
200
300
400
500

```
N--;
```

Dely Gil

Parte I. Operaciones con Arreglos Unidimensionales

1. Se tiene un Arreglo lineal de N números. Hallar la suma de los N números. (N es ingresado por el Usuario).
2. Se tienen N empleados en una compañía y se ha ideado llenar un arreglo lineal A con los sueldos de los empleados, un arreglo B con las asignaciones totales de cada empleado, un arreglo C con las deducciones de cada uno. Crear un arreglo T que contenga el neto a pagar a cada empleado. (Neto a pagar = sueldo + asignaciones – deducciones)
3. Dado un arreglo lineal de números, sumar separadamente los números pares y los números impares.
4. Se tiene dos arreglos unidimensionales que guarda las edades de un grupo de personas, se pide hallar el mayor valor.
5. Se tienen tres arreglos unidimensionales que guardan los precios unitarios (PU) las cantidades compradas (CC) y la descripción respectivamente de N productos distintos. Se pide:
 - a. Crear un arreglo con el total gastado en compras por producto (TG= PU*CC)
 - b. Calcular el total general de todas las compras.
 - c. Mostrar la descripción y el total del producto que obtuvo mayor gasto.
6. Una Agencia administradora de inmuebles ha decidido guardar en un arreglo lineal de N posiciones los alquileres que cobran mensualmente a N viviendas que actualmente administran. En otro arreglo de igual número de posiciones guardan los porcentajes de ganancias por cada vivienda. Crear un nuevo arreglo con las ganancias por cada vivienda.
7. Una empresa decide guardar en un arreglo lineal el número de ventas realizadas en un mes por cada vendedor. Indicar cuántos de ellos hicieron igual número de ventas.
8. Codificar un programa que lea un vector de *MaxLista* elementos de tipo cadena(Nombres de personas(validar)), los guarde en un arreglo y permita consultarlos de acuerdo con el número de posición que ocupa dicho nombre en el vector. Realizar procedimientos para leer el vector, validar los nombres y buscar el nombre en la posición.
9. En el ejercicio anterior se desea hacer un procedimiento que inserte un elemento dentro del array en una posición determinada que pasamos como parámetro, respetando el orden que tenía, siempre y cuando haya sitio para hacer la operación.
10. En el ejercicio 10, se desea diseñar un procedimiento que elimine un elemento dentro del array en una posición que pasamos como parámetro, conservando el array en el mismo orden.
11. Elaborar un programa que lea 30 números y que imprima el número mayor, menor y el número de veces que se repiten ambos. Debe utilizar la estructura de dato VECTOR.
12. Elaborar un programa que lea un arreglo de 15 números , que pregunte si se desea introducir un nuevo número en lugar de los cualesquiera que están en el arreglo; entonces leer el número a introducir y el lugar del elemento por el cual se cambiará. Imprimir el arreglo antes y después del cambio. Realizar este ejercicio con procedimientos : Leer_Vector, Cambio_Vector e Imprimir_Vectores.
13. Codifique un programa tal, que dado como entrada un arreglo unidimensional de enteros y un número entero, determine cuántas veces se encuentra este número dentro del arreglo.
14. Escribir un procedimiento que acepte como parámetro un vector que pueda contener elementos duplicados. El procedimiento debe sustituir cada valor repetido por -5 y devolver el vector modificado y el número de entradas modificadas.
15. Realice un programa que reciba como entrada un arreglo unidimensional desordenado de enteros (posiblemente repetidos) y genere como salida ese mismo vector, pero sin repeticiones.
16. Realice un programa que reciba como entrada un arreglo unidimensional ordenado de enteros (posiblemente repetidos) y genere como salida una lista con los números enteros, pero sin repeticiones.

Parte II. Operaciones con Arreglos Bidimensionales

1. Sumar los elementos de cada fila y cada columna de una matriz
2. Realice un programa que calcule la tabla de multiplicar del 1 al 10 almacenado los valores en una tabla. Imprimir dicha tabla.
3. Codificar un programa que genere una matriz n x m , en la cual asigne ceros a todos los elementos, excepto a los de la diagonal principal donde asignaran unos.
4. Codificar un programa que genere una matriz 10 x 10 con ceros en la diagonal principal hacia arriba.
5. Dada una matriz de M*M elementos, hacer un programa que construya un vector B, donde cada uno de sus componentes sea la suma de los elementos de valores numéricos pares de las filas de la matriz.
6. Escribir un programa que lea las dimensiones de una matriz, lea y visualice la matriz y a continuación encuentre el mayor y menor elemento de la matriz y sus posiciones.

7. Una matriz se dice que es simétrica si $A(i,j) = A(j,i)$ para todo i,j dentro de los límites de las matriz. Escribir un subprograma que indique si una matriz dada es simétrica o no simétrica.
8. Determinar si una matriz de tres filas y tres columnas es un cuadrado mágico. se considera un cuadrado mágico aquel en el cual las filas, columnas y las diagonales principal y secundaria suman los mismo.
9. Dada una matriz de orden $N \times M$, hallar el elemento de mayor valor absoluto de cada fila y almacenarlo en la última columna o en otro vector.
10. Dada una matriz de orden $N \times M$ de elementos no repetidos, elabore un programa que encuentre los 2 elementos de mayor valor absoluto almacenándolos en vectores paralelos . Imprimir dichos valores, mediante un subprograma.
11. Hacer un programa que a partir de dos arreglos A,B de orden $N \times M$ y orden $N' \times M'$ respectivamente obtenga un arreglo C unidimensional donde cada elemento de C será igual al valor promedio de los elementos de la fila de A, multiplicado por el menor elemento de la columna de B elevado a la posición que ocupa. Imprimir al final A,B,C.
12. Dada una matriz cuadrada de orden N, construir un programa que obtenga otra matriz de orden $2N$ que contenga a la primera matriz como submatriz diagonal, siendo nulos los elementos restantes de la matriz.

$$A = \begin{vmatrix} A & B & C \\ D & E & F \\ G & H & I \end{vmatrix} \quad B = \begin{vmatrix} A & B & C & 0 & 0 & 0 \\ D & E & F & 0 & 0 & 0 \\ G & H & I & 0 & 0 & 0 \\ 0 & 0 & 0 & A & B & C \\ 0 & 0 & 0 & D & E & F \\ 0 & 0 & 0 & G & H & I \end{vmatrix}$$

Parte III.

1. Se tienen los nombres y los sueldos de los trabajadores de una empresa. Se desea saber cuántos y quiénes tienen un sueldo superior al promedio.
2. El Departamento de personal de una escuela tiene información sobre nombre, edad y sexo de cada uno de los profesores adscritos al mismo. Escriba un programa que calcule (Proceso) e imprima (Salida) lo siguiente:
 - a. Edad promedio del grupo de profesores.
 - b. Nombre del profesor más joven del grupo
 - c. Nombre del profesor con mayor edad
 - d. Número de profesoras con edad mayor al promedio
 - e. Número de profesores con edad menor al promedio.
3. Se tiene la producción total de toneladas de cereales(arroz, avena,cebada,trigo) cosechadas durante cada mes del año anterior. Elaborar un programa que proporcione la siguiente información:
 - a. El promedio anual de toneladas cosechadas.
 - b. ¿Cuántos meses tuvieron una cosecha superior al promedio anual?
 - c. ¿Cuántos meses tuvieron una cosecha inferior al promedio anual?
 - d. ¿Cuál fue el mes en el que se produjeron mayor número de toneladas?
4. Se tiene la producción de los siete días de la semana de 20 plantas. Elaborar un programa que lea estos datos e imprima el número de la planta que tuvo la mayor producción semanal.
5. Igual que el ejercicio anterior, sólo que ahora deberá imprimir el número de planta con mayor producción en un día específico (dato que ingresa el usuario), y con cuánta producción.
6. En una papelería se maneja información sobre los 8 modelos diferentes de cuadernos que venden. Por cada modelo se tiene: la distribuidora (Norma, Caribe, Alpes), el modelo (A, B, C, D, E, F, G, H) y el precio. Realice un programa que calcule lo siguiente:
 - a. El total recaudado por modelo a lo largo de los 30 días.
 - b. El total recaudado por día.
 - c. ¿Cuál fue el modelo que más dinero produjo en los 30 días?
7. Se tienen los costos de producción de tres departamentos (dulces, bebidas y conservas), correspondientes a los 12 meses del año anterior. Elabore un programa que proporcione la siguiente información:
 - a. ¿En qué mes se registró el mayor costo de producción de dulces?
 - b. Promedio anual de los costos de producción e bebidas
 - c. ¿En qué mes se registró el mayor costo de producción en bebidas, y en qué mes el menor costo?
 - d. ¿Cuál fue el rubro que tuvo el menor costo de producción en diciembre?
8. Se tienen las temperaturas promedio diarias del año anterior de la ciudad de Mérida, almacenadas en una tabla TEMP de 12 filas x 31 columnas. Elabore un programa que proporcione la siguiente información:
 - a. La temperatura más alta registrada en el año anterior, y cuál fue el día y mes en que se registró.
 - b. El mes que tuvo el promedio de temperaturas más alto.
 - c. El promedio mensual de temperaturas

9. Una empresa automotriz necesita un programa para manejar los montos de ventas de sus 4 Sucursales (Cabriales, Naguanagua, Michelena, San Diego) a lo largo de los últimos M años. La información que necesitan los directores de la empresa, para tomar decisiones es la siguiente:
- Sucursal que más ha vendido en los M años.
 - Promedio de ventas por año.
 - Año con mayor promedio de ventas.
10. En la U.L.A. (Universidad de Los Andes) se conoce el número de alumnos que ingresaron en sus 4 diferentes carreras de Ingeniería (Ingeniería Civil, Mecánica, Eléctrica y Sistemas), en los últimos 5 años. Construya un programa que proporcione la siguiente información:
- Total de alumnos por año.
 - Porcentaje de alumnos ingresados en el año X de la carrera Y.
 - ¿En qué año y en qué carrera se dio el menor ingreso de alumnos?
 - Año en el cual la carrera Y tuvo el mayor ingreso de alumnos
11. La Administración de una compañía quiere conocer el ausentismo registrado en cada uno de los 12 departamentos (Administración, Recursos Humanos, Informática, Contabilidad, Control de Calidad, Costos, Compras, Almacén, Ventas, Nómina, Planta, Seguridad Industrial) que tiene la empresa en un mes laboral. Para ello se requiere elaborar un programa que determine:
- Total ausentismo mensual por los departamentos.
 - Total ausentismo por día en la empresa
 - Departamento con mayor y menor ausentismo en el mes.
12. Realice un programa que permita manejar la venta de localidades de un teatro, en donde las filas están marcadas de la 'A' a la 'Z' y cada fila tiene asientos marcados del 1 al 10. Se debe pedir al usuario la elección de asientos concretos hasta que escoja alguno que no esté reservado; posteriormente pedirá el nombre del cliente para asignarle su lugar.
13. Una compañía almacena a información relacionada a sus proveedores (1..100): nombre, ciudad donde vive y el artículo que provee (1..50). La información del proveedor i y el artículo j se almacenará en una tabla de tipo boolean. Si el proveedor i provee el artículo j, entonces se almacena verdadero de lo contrario almacena falso. Construya un sistema que permita llevar las siguientes transacciones:
- Dado el nombre del proveedor, informar el nombre de la ciudad donde vive y el número de artículos que provee.
 - Actualizar le nombre de la ciudad, en caso de que el proveedor cambie de domicilio: Los datos serán en este caso, el nombre del proveedor y el nombre de la ciudad a la cual se mudó.
 - La compañía incorpora si es posible, recuerde que el límite es 100, un nuevo proveedor: Actualizar los arreglos correspondientes.
 - La compañía da de baja a un proveedor.
14. Se tiene 4 arreglos que contiene la siguiente información: PAGOS[i,j] representa el pago que hizo el cliente i en el día j de la semana(1..5); COMPRA[i,j] representa la compra que hizo el cliente i en el día j; NOMBRES[i] expresa el nombre del cliente i; SALDOS[i] representa el saldo del cliente i. Construya un sistema que pueda efectuar las siguientes operaciones:
- Actualizar los SALDOS. Para ésto deberá sumar al saldo actual del cliente las compras que hizo durante la semana y restarle también los pagos que hizo durante la semana.
 - Obtener el día de la semana en que se hicieron los pagos.
 - Indicar el nombre del cliente que hizo más compras durante la semana
15. Escribir un programa que ingrese las últimas elecciones a alcalde del pueblo x, en la cual se tienen 4 candidatos y 5 distritos. Se desea lo siguiente:
- Imprimir la tabla con los datos y sus cabeceras incluidas.
 - Calcular e imprimir el número total de votos recibidos por cada candidato y el porcentaje del total de votos por cada candidato y el porcentaje de total de votos emitidos.
 - Si algún candidato recibe más del 50% de los datos, el programa imprimirá un mensaje declarándolo ganador.
 - Si ningún candidato recibe más del 50% de lo datos, el programa debe imprimir el nombre de los candidatos más votados, que serán los que pasen a la segunda ronda de las elecciones.
16. Una agencia de viajes de vehículos automóviles distribuye quince modelos diferentes y tiene en su planilla diez vendedores. Se desea un programa que escriba un informe mensual de las ventas por vendedor y modelo, el cual debe calcular el número de automóviles vendidos por cada vendedor y el número total de cada modelo vendido por todos los vendedores. Así mismo para entregar al mejor vendedor, necesita saber cuál es el vendedor que más coches ha vendido.
17. Se tiene un coro constituido por cuatro voces, a saber: SOPRANO, CONTRA-ALTO, TENOR y BARÍTONO. Para un concierto de 5 piezas se tiene la siguiente información almacenada en un arreglo llamado CORO donde:
- $$\text{CORO}(i,j) = 1 \text{ si la voz } i \text{ interviene en la pieza } j.$$
- $$\text{CORO}(i,j) = 0 \text{ si la voz no interviene en la pieza } j.$$
- Se desea determinar:
- Cuales voces intervienen más veces en el concierto.
 - Cuales piezas tienen un solista y a cual voz corresponde.
 - Cual voz interpreta el mayor número de piezas consecutivas.
18. Una empresa de venta de productos farmacéuticos desea realizar una estadística de las ventas realizadas de cada uno de los productos a lo largo del año. Distribuye un total de 10 productos, por lo tanto se almacenará en una tabla de 10 filas por 12 columnas. Se desea conocer:
- El total de ventas de cada uno de los productos.
 - El total de ventas de cada mes
 - El producto más vendido en cada mes, almacenar en dos vectores paralelos o en un registro la siguiente información: la venta del mes y el mes correspondiente de la venta.
 - La descripción, el mes y la cantidad del producto más vendido.

Como resultado final, se desea realizar un listado con el siguiente formato:

	Enero	Febrero	...	Diciembre	Total Producto
Producto 1					
...					
Producto 10					

Total Mes					
Producto más vendido					

Nombre del producto y mes del producto más vendido en cualquier mes del año.					

Parte IV. Ordenamiento, Búsqueda.

1. Codificar un programa con dos procedimientos, uno para leer el vector y el otro para ordenarlos de mayor a menor.
2. Dados dos arreglos A y B, con elementos ordenados en forma ascendente, se quiere crear un arreglo C ordenado en forma descendente con los elementos de A y B.
3. Dado un arreglo A con n elementos enteros ordenados en forma ascendente y un entero X, se quiere determinar si dicho entero pertenece al arreglo. En caso afirmativo indique la posición que ocupa x en el arreglo, en caso negativo emitir un mensaje. Use el algoritmo de búsqueda con centinela y el algoritmo de búsqueda binaria.
4. Simule por medio de un programa, un directorio telefónico que lea nombres y números telefónicos en dos arreglos paralelos. Los nombres están ordenados alfabéticamente en orden ascendente. Cuando el usuario teclee el nombre, el programa deberá verificar si éste ya está en el directorio, en cuyo caso imprimirá el número telefónico. De no ser así, y si hay capacidad, pedirá al usuario que escriba el teléfono y agregará nombre y número a los arreglos.

Parte IV. Mezcla

1. Se tienen dos arreglos: CINES y TEATROS. El primero almacena los nombres de todos los cines de la ciudad. Está ordenado alfabéticamente de manera ascendente. El segundo arreglo guarda los nombres de todos los teatros de la ciudad, y está ordenado alfabéticamente de manera descendente. Construya un programa que mezcle estos arreglos formando un tercero, ENTRETENIMIENTOS, de tal manera que quede alfabéticamente de manera ascendente.
2. Se tiene tres arreglos: SUR; CENTRO y NORTE que almacene los nombres de los países del Sur, Centro y Norteamérica, respectivamente. Los tres arreglos están ordenados alfabéticamente. Construya un programa que mezcle los tres arreglos anteriores, formando un cuarto arreglos AMERICA, en el cual aparezcan los nombres de todos los países del continente ordenados alfabéticamente.
3. Se tiene un arreglo NUM de una dimensión y N elementos almacenados aleatoriamente. Elabore un programa que permita llevar otro arreglo POS_ORDEN el cual contiene el orden de menor a mayor que debería ocupar cada elemento si se ordenara el vector NUM. Ejemplo:

Posición	1	2	3	4	5	6	7	8	9	▪ El 1 ^{er} elemento del arreglo NUM(1) = 10; ocuparía la posición 5.
NUM	10	6	14	16	9	7	12	1	15	POS_ORDEN(1) = 5
POS_ORDEN	5	2	7	9	4	3	6	1	8	El 2 ^{do} elemento, NUM(2) = 6 ocuparía la posición 2. POS_ORDEN(2) = 2
										▪ El 4 ^{to} elemento NUM(4) = 16 ocuparía la posición 9

¡VENGO PRONTO!
Y MI GALARDÓN CONMIGO,
PARA ARECOMPENSAR A CADA UNO SEGÚN SEA SU OBRA.
YO SOY EL ALFA Y EL OMEGA,
EL PRINCIPIO Y EL FIN ,
EL PRIMERO Y EL ÚLTIMO.
Apocalipsis 22:12-13