



## GUIA TEORICA N° 3 HERRAMIENTAS DE INFORMÁTICA 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, Marzo, 2008

### 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.

### 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];
}
```

**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.

**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]);
  }
}
```

**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}
```

```
    }
```

```
}
```

**ARREGLOS PARALELOS**

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

Nombres[0]

Nombres[1]

Nombres[n-1]

<b>Job</b>
<b>Josue</b>
.
.
.
<b>Isaias</b>

Edades[0]

Edades[1]

Edades[n-1]

<b>82</b>
<b>28</b>
.
.
.
<b>9</b>

```

#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;
}

```

```

#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");
    }
}

```

**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;
}
```

```
//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++;
    }
}
```

**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;
}
```

```
#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]);
    }
}
```

```

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 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);
}

```

```

/*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

```

/\*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]	<b>Job</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>
Alumnos[1]	<b>Josue</b>	<b>20</b>	<b>20</b>	<b>20</b>	<b>20</b>
Alumnos[2]	<b>Luis</b>	<b>15</b>	<b>15</b>	<b>15</b>	<b>15</b>
Alumnos[3]	<b>Isaias</b>	<b>05</b>	<b>05</b>	<b>05</b>	<b>05</b>

```
#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]);
    }
}
```

```
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.**

```

//Promedio por asignatura
for(j=0;j<N;j++){
    suma =0;
    for(i=0;i<M;i++)
        suma += Exámenes[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 (Exámenes[i][j]>=may){
            may = Exámenes[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;
}

```

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 Países[P][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;

```

```
//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]);
}
}
```

```
//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;
}
```

### MANEJO DE MATRICES

Diagonal Principal ( $i = j$ )

A11	A12	A13	A14	A15	A16
A21	A22	A23	A24	A25	A26
A31	A32	A33	A34	A35	A36
A41	A42	A43	A44	A45	A46
A51	A52	A53	A54	A55	A56
A61	A62	A63	A64	A65	A66

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

### MANEJO DE MATRICES

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

A11	A12	A13	A14	A15	A16
A21	A22	A23	A24	A25	A26
A31	A32	A33	A34	A35	A36
A41	A42	A43	A44	A45	A46
A51	A52	A53	A54	A55	A56
A61	A62	A63	A64	A65	A66

```
for(i=0;i<N;i++) {
  J = N - i + 1;
  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;
```

### MANEJO DE MATRICES

Diagonal Principal Superior ( $i < j$ )

A11	A12	A13	A14	A15	A16
A21	A22	A23	A24	A25	A26
A31	A32	A33	A34	A35	A36
A41	A42	A43	A44	A45	A46
A51	A52	A53	A54	A55	A56
A61	A62	A63	A64	A65	A66

```
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;
```

### MANEJO DE MATRICES

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

A11	A12	A13	A14	A15	A16
A21	A22	A23	A24	A25	A26
A31	A32	A33	A34	A35	A36
A41	A42	A43	A44	A45	A46
A51	A52	A53	A54	A55	A56
A61	A62	A63	A64	A65	A66

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

**MANEJO DE MATRICES**  
Diagonal Principal Inferior ( $i > j$ )

A11	A12	A13	A14	A15	A16
A21	A22	A23	A24	A25	A26
A31	A32	A33	A34	A35	A36
A41	A42	A43	A44	A45	A46
A51	A52	A53	A54	A55	A56
A61	A62	A63	A64	A65	A66

```
for(i=1;i<N;i++)
  for(j=1;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;
```

**MANEJO DE MATRICES**  
Diagonal Secundaria Inferior ( $i+j > N+1$ )

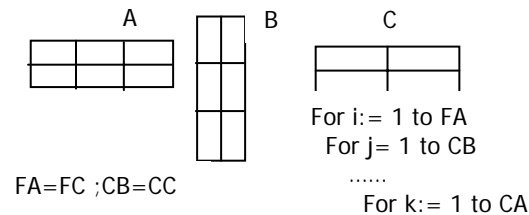
A11	A12	A13	A14	A15	A16
A21	A22	A23	A24	A25	A26
A31	A32	A33	A34	A35	A36
A41	A42	A43	A44	A45	A46
A51	A52	A53	A54	A55	A56
A61	A62	A63	A64	A65	A66

```
for(i=1;i<N;i++)
  for(j=N-i+2;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;
```

**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];
  }
```



**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)