

# Ejercicios Resueltos - Matrices

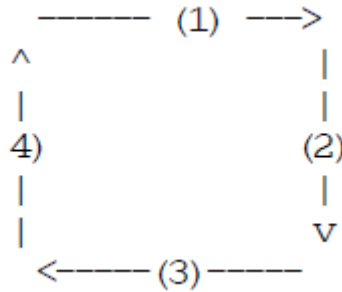
- 1) Multiplicar (si es posible) dos matrices A y B de tamaños  $m \times n$  y  $p \times q$  respectivamente (siendo estos últimos números naturales). Guardar el resultado en una matriz C e imprimirla en pantalla.

```
#include<stdio.h>

int main(){
    int m, n, p, q, i, j, k;
    printf("Ingrese la cantidad de filas y columnas de la matriz A:");
    printf("\nFilas: "); scanf("%d",&m);
    printf("Columnas: "); scanf("%d",&n);
    printf("\n\nIngrese la cantidad de filas y columnas de la matriz B:");
    printf("\nFilas: "); scanf("%d",&p);
    printf("Columnas: "); scanf("%d",&q);
    if (n!=p){ printf("\n\nLas matrices no pueden multiplicarse\n"); return 0;}
    int A[m][n], B[p][q], C[m][q];
    //Lectura de datos
    printf("\n\nIngrese los datos de la matriz A:\n");
    for (i=0; i<m; i++){
        for(j=0; j<n; j++){
            printf("A[%d][%d]: ", i, j); scanf("%d",&A[i][j]);
        }
    }
    printf("\n\nIngrese los datos de la matriz B:\n");
    for(i=0; i<p; i++){
        for(j=0; j<q; j++){
            printf("B[%d][%d]: ", i, j); scanf("%d",&B[i][j]);
        }
    }
    //Proceso
    for(i=0; i<m; i++){
        for(j=0; j<q; j++){
            C[i][j]=0;
            for(k=0; k<n; k++){
                C[i][j] = C[i][j] + A[i][k] * B[k][j];
            }
        }
    }
    //Impresion de resultados
    printf("\n\nMatriz C:\n");
    for(i=0; i<m; i++){
        for(j=0; j<q; j++){
            printf("%d\t", C[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

- 2) Dadas la cantidad de filas y columnas máximas, generar una matriz "caracol"

1	2	3	4	5
12	13	14	15	6
11	10	9	8	7



```
#include<stdio.h>

int main(){
    int m, n, i, j, k, fil, col, filmin, filmax, colmin, colmax, dir, cont=0;
    printf("Ingrese la cantidad de filas y columnas de la matriz C:");
    printf("\nFilas: "); scanf("%d",&m);
    printf("Columnas: "); scanf("%d",&n);
    int C[m][n];
    dir=1; filmin=0; filmax=m-1; colmin=0; colmax=n-1; fil=0; col=0;
    for(k=1; k<=(m*n); k++) {
        C[fil][col] = k;
        if (dir == 1) {
            col++;
            if (col == colmax) {
                dir = 2;
                filmin++;
            }
        } else if (dir == 2) {
            fil++;
            if (fil == filmax) {
                dir = 3;
                colmax--;
            }
        } else if (dir == 3){
            col--;
            if (col == colmin){
                dir = 4;
                filmax--;
            }
        } else {
            fil--;
            if (fil == filmin){
                dir = 1;
                colmin++;
            }
        }
    }

    printf("\n\nMatriz C:\n");
    for(i=0; i<m; i++){
        for(j=0; j<n; j++){
            printf("%d\t",C[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

- 3) Ordenar alfabéticamente un conjunto de  $n$  palabras que se representan mediante una matriz de caracteres. Cada palabra es una fila de la matriz y no supera los 20 caracteres.

E	S	T	A	\0		
E	S	\0				
U	N	A	\0	\0		
P	R	U	E	B	A	\0

```
#include<stdio.h>

int comparar(char cad1[], char cad2[]){
    int i=0, a;
    while((cad1[i] != '\0') && (cad1[i] == cad2[i])) i++;
    a = cad1[i] - cad2[i];
    return a;
}

void copia(char cad1[], char cad2[]){
    int i=0;
    while (cad2[i] != '\0') {
        cad1[i] = cad2[i];
        i++;
    }
    cad1[i]='\0';
}

int main(){
    int i,n;
    printf("Ingrese la cantidad de palabras: "); scanf("%d",&n);
    char pal[n][21], cad[21];
    printf("\nIngrese las palabras:\n");
    for(i=0; i<n; i++){
        printf("Palabra %d: ", (i+1));
        scanf("%s", pal[i]);
    }
    //Proceso
    int band=1, j=1, a;
    while (band){
        band=0;
        for(i=0; i<(n-j); i++){
            a = comparar(pal[i], pal[i+1]);
            if (a > 0) {
                band = 1;
                copia(cad, pal[i]);
                copia(pal[i], pal[i+1]);
                copia(pal[i+1], cad);
            }
        }
        j++;
    }

    printf("\nPalabras ordenadas:\n");
    for(i=0; i<n; i++){
        puts(pal[i]);
    }

    return 0;
}
```