

# SOLUCIÓN DEL EXAMEN DEL 2 DE JUNIO DE 2003

—Parte común—

## CUESTIÓN 1:

1. La velocidad de acceso, en la MC es del orden de los nanosegundos y en los discos duros de los milisegundos. La velocidad de trabajo de la MC es 3 órdenes de magnitud superior a la del disco duro.
2. Los discos duros no necesitan alimentación eléctrica para mantener la información que en ellos se almacena.

## CUESTIÓN 2:

*True Color* o 24 bits de color implica 3 bytes u octetos por punto o *pixel* de pantalla, por tanto:  
 $1280 * 1024 * 3 = 3.75$  MBytes

## PROBLEMA 1:

```
#include <stdio.h>

#define N 3
#define M 4

int main()
{
    int    A[N][M];
    int    i, j, maximo, fila, columna;

    for( i=0; i < N; i++ ) {
        for( j=0; j < M; j++ ) {
            printf( "\n A[%d][%d]: ", i, j );
            scanf( "%d", &A[i][j] );
        }
    }

    fila = columna = 0;
    maximo = A[fila][columna];

    for( i=0; i < N; i++ ) {
        for( j=0; j < M; j++ ) {
            if ( A[i][j] > maximo ) {
                fila = i;
                columna = j;
                maximo = A[fila][columna];
            }
        }
    }

    printf( "\n El valor máximo ha sido localizado " );
    printf( "en la columna %d de la fila %d\n", columna, fila );
}
```

```
    return 0;
}
```

## PROBLEMA 2:

```
#include <stdio.h>

#define N    10

struct premio {
    char    nombre[30], premiado[30];
    float    dotacion;
};

int menu()
{
    int        opc;

    printf( "\n\n MENÚ \n\n" );

    printf( " 1) Entrada de datos\n" );
    printf( " 2) Listado completo\n" );
    printf( " 3) Datos premio de mayor cuantía\n" );
    printf( " 0) Salir\n" );

    printf( "\n\n Escoge una opción: " );
    scanf( "%d", &opc );

    return opc;
}

void entrada_de_datos( struct premio P[] );
void listado( struct premio P[] );
void premio_mas_alto( struct premio P[] );

int main()
{
    struct premio    P[N];
    int                opcion;

    do {
        opcion = menu();
        switch( opcion ) {
            case 1 : entrada_de_datos( P ); break;
            case 2 : listado( P ); break;
            case 3 : premio_mas_alto( P ); break;
        }
    } while( opcion != 0 );

    return 0;
}
```

```

void entrada_de_datos( struct premio P[] )
{
    int      i;

    for( i=0; i < N; i++ ) {
        printf( "\n Nombre del premio.: " ); gets( P[i].nombre );
        printf( "\n Nombre del premiado: " ); gets( P[i].premiado );
        printf( "\n Dotación del premio: " ); scanf( "%f", &P[i].dotacion );
    }
}

void listado( struct premio P[] )
{
    int      i;

    printf( "\n %-20.20s  %-25.25s  %-12.12s\n",
            "PREMIO", "PREMIADO", "DOTACIÓN" );
    for( i=0; i < N; i++ ) {
        printf( " %-20.20s  %-25.25s  %12.2f\n",
            P[i].nombre, P[i].premiado, P[i].dotacion );
    }
}

void premio_mas_alto( struct premio P[] )
{
    int      i, imax;
    float    max_dot;

    imax = 0;
    max_dot = P[imax].dotacion;

    for( i=1; i < N; i++ ) {
        if ( P[i].dotacion > max_dot ) {
            max_dot = P[i].dotacion;
            imax = i;
        }
    }

    printf( "\n Los datos del premio más alto son: " );
    printf( " %s \n", P[imax].nombre );
    printf( " %s \n", P[imax].premiado );
    printf( " %10.2f\n", P[imax].dotacion );
}

```

—Parte de prácticas—

PROBLEMA 1:

```

struct fecha { int d,m,a; };

struct coche {
    char          nombre[25], apellidos[25];
    struct fecha  fnac, fcontrato;
    int           num_competiciones, num_victorias;
    float         potencia;
    struct fecha  fmontaje;
}

```

```
        int          num_reparaciones;
    };
```

## PROBLEMA 2: (TRAZA)

0	4
1	4
2	8
3	8
4	16
5	16

## PROBLEMA 3:

```
#include <stdio.h>

int cuenta_palabras( char str[] )
{
    int    i, c;

    if ( str[0] == '\0' ) return 0;

    c=0;
    if ( str[0] != ' ' ) c++;
    for( i=1; str[i]!='\0'; i++ ) {
        c += ( str[i-1] == ' ' && str[i] != ' ' );
    }

    return c;
}

int main()
{
    char    frase[256];
    int     num_pal;

    do {
        printf( "\n Dame una frase: " );
        gets( frase );
        num_pal = cuenta_palabras( frase );
        printf( "\n %d palabras\n", num_pal );
    } while( num_pal > 0 );

    return 0;
}
```