

# SOLUCIÓN DEL EXAMEN DEL 30 DE ENERO DE 2003

—Parte común—

## CUESTIONES:

1)

```
t=0.0;
while( t < 100.0 ) {
    printf( " %.3f\n", t*t );
    t += 0.5;
}
```

2)

9 bits, porque 2 elevado a 9 son 512, de manera que con 9 bits sobran combinaciones, y con 8 faltan, ya que 2 elevado 8 da 256.

## PROBLEMA 1:

```
#include <stdio.h>

#define M    1000
#define N    1200

int main()
{
    int          A[M][N], B[M][N+M-1], i, j;

    /* Lectura matriz A */
    for( i=0; i < M; i++ ) {
        for( j=0; j < N; j++ ) {
            printf( "A[%d][%d] ", i, j );
            scanf( "%d", &A[i][j] );
        }
    }

    /* Paso de A a B */
    for( i=0; i < M; i++ ) {
        for( j=0; j < i; j++ ) B[i][j] = 0;
        for( j=0; j < N; j++ ) B[i][j+i] = A[i][j];
        for( j=N+i; j < M+N-1; j++ ) B[i][j] = 0;
    }

    /* Muestra del resultado */
    for( i=0; i < M; i++ ) {
        for( j=0; j < N+M-1; j++ ) {
            printf( "B[%d][%d] = %5d\n", i, j, B[i][j] );
        }
    }

    return 0;
}
```

## PROBLEMA 2:

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

```
#define N 1100
```

```
struct alumno {  
    int grupo;  
    char nombre[32];  
    double parciales[6];  
};
```

1)

```
struct alumno FIN[N];
```

2)

```
void nota_media( struct alumno V[] )  
{  
    int i, j;  
    double total, minima;  
  
    for( i=0; i < N; i++ ) {  
  
        minima = V[i].parciales[0];  
        total = 0.0;  
        for( j=0; j < 5; j++ ) {  
            total += V[i].parciales[j];  
            if ( V[i].parciales[j] < minima )  
                minima = V[i].parciales[j];  
        }  
        total = total - minima;  
  
        V[i].parciales[5] = total / 4.0;  
    }  
}
```

3)

```
void listado( struct alumno V[] )  
{  
    int i;  
  
    for( i=0; i < N; i++ ) {  
        if ( V[i].parciales[5] >= 4.0 ) {  
            printf( " %4d %-30.30s %.1f\n",  
                V[i].grupo, V[i].nombre, V[i].parciales[5] );  
        }  
    }  
}
```

—Parte de prácticas—

PROBLEMA 1: versión con vectores

```
#include <stdio.h>

#define    N    3

int main()
{
    int        maximos[N], valor, i, auxiliar, candidato,
              seguir_comprobando;

    for( i=0; i < N; i++ ) maximos[i] = -1;

    do {
        /* Pido un número al usuario */
        printf( "\n Dame un valor entero: " );
        scanf( "%d", &valor );

        /* Voy comprobando en el vector de máximos si hay valores inferiores,
           en cuyo caso los intercambio */
        candidato = valor;
        seguir_comprobando = 1;
        for( i=0; i < N && seguir_comprobando; i++ ) {
            if ( candidato > maximos[i] ) {
                /* Intercambio entre candidato y maximos[i] */
                auxiliar = candidato;
                candidato = maximos[i];
                maximos[i] = auxiliar;
            } else if ( candidato == maximos[i] )
                seguir_comprobando=0;
        }
    } while( valor >= 0 );

    /* Muestro el resultado. */
    printf( "\n Máximos: " );
    for( i=0; i < N; i++ ) {
        if ( maximos[i] >= 0 ) printf( " %d ", maximos[i] );
    }
    printf( "\n" );

    return 0;
}
```

PROBLEMA 1: versión sin vectores

```
#include <stdio.h>

int main()
{
    int        max1, max2, max3, valor, i, aux, candidato;
```

```

max1 = max2 = max3 = -1;
do {
    /* Pido un número al usuario */
    printf( "\n Dame un valor entero: " );
    scanf( "%d", &valor );

    candidato=valor;
    if ( candidato > max1 ) {
        aux = max1;    max1 = candidato; candidato = aux;
    }
    if ( candidato > max2 && candidato != max1 ) {
        aux = max2;    max2 = candidato; candidato = aux;
    }
    if ( candidato > max3 && candidato != max1 && candidato != max2 ) {
        aux = max3;    max3 = candidato; candidato = aux;
    }

} while( valor >= 0 );

/* Muestro el resultado. */
printf( "\n Máximos: " );
if ( max1 >= 0 ) printf( " %d ", max1 );
if ( max2 >= 0 ) printf( " %d ", max2 );
if ( max3 >= 0 ) printf( " %d ", max3 );
printf( "\n" );

return 0;
}

```

## PROBLEMA 2:

```

struct fecha {
    int      d, m, a;
};
struct averia {
    char      persona_contacto[32], extension[6];
    char      edificio[20], ubicacion[12];
    char      descripcion[40], comentario[60];
    int       codigo;
    struct fecha favisos, fresolucion;
    char      estado;
};

```

## TRAZA

```

0 0
1 1
2 2
3 6
4 12
5 20

```