
Práctica: Arrays

Ejemplo 1: frecuencia de ocurrencia dígitos

```
/******\
* Programa: freq_ocurr_digitos.c *
* Descripción: Prog. que cuenta la frecuencia de ocurrencia de digitos *
* Autor: Pedro Corcuera *
* Revisión: 1.0 2/02/2008 *
\*****/
#include <stdio.h>

main()
{
    int c,i;
    int ndigit[10] = { 0};    /* Inicializacion express del array */

    printf ("Escribe numeros y pulsa Enter. Para terminar Ctrl-Z Enter\n");
    while ((c=getchar())!=EOF)
        if (c>='0' && c<='9') /* Si c es un digito */
            ++ndigit[c-'0']; /* incrementa el contador del digito c-'0' */

    printf("Frecuencia de ocurrencia de digitos \n");
    for (i=0; i<10 ; ++i)
        printf(" %d = %d\n", i, ndigit[i]);

    return 0;
}
```

Ejemplo 2: adivina números

```
/******\
* Programa: busq_binaria.c *
* Descripción: Prog. que genera N numeros random y permite que el *
* usuario adivine si esta un numero. Ordena por burbuja *
* y usa busqueda binaria *
\*****/
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define N 10

main()
{
    int lista[N], num, encontrado;
    int i, j, t;
    int izq, der, mitad;

    /* Crea lista de valores random entre 0 y 100.
       rand() devuelve un numero random entre 0 y RAND_MAX */
    srand( (unsigned)time( NULL ) ); /* semilla para random */
    for( i = 0; i < N; i++ )
        lista[i] = (float)rand()/RAND_MAX*100;
```

Ejemplo 2: adivina números

```
/* Ordenacion por metodo de burbuja para aplicar busqueda binaria */
for(i = N-1; i > 0 ; i--)
    for(j = 0 ; j < i ; j++)
        if (lista[j] > lista[j+1])
            { t = lista[j]; lista[j] = lista[j+1]; lista[j+1] = t; }
printf("He generado una lista de 10 numeros enteros entre 0 y 100.\n \
Adivina los numeros de la lista (Para terminar pulsar tecla no digito)\n");
while (scanf(" %d", &num)==1)
{
    izq = 0; der = N-1; encontrado = 0;
    while (izq <= der)
    {
        mitad = (izq + der)/2;
        if (num < lista[mitad]) der = mitad - 1;
        else if (num > lista[mitad]) izq = mitad + 1;
        else {
            encontrado = 1; printf(" %d si esta\n", num); /* elem. en lista */
            break;
        }
    }
    if (encontrado == 0) printf(" %d no esta\n", num); /* no encontrado */
}
return 0;
}
```

Ejemplo1: frecuencia de ocurrencia letras

```
/******\
* Programa: frecuencia_letras.c *
* Descripción: Prog. que cuenta la frecuencia de ocurrencia de letras *
\*****/
#include <stdio.h>
#include <ctype.h>
#define N_LETRAS 27 /* incluye la ñ */

main()
{
    int c,i, nletras[N_LETRAS] = { 0};

    printf ("Introduce texto y pulsa Enter. Para terminar Ctrl-Z Enter\n");
    while ((c=getchar())!=EOF)
        if (isalpha(c)) {
            if (isupper(c)) c = tolower(c);
            ++nletras[c-'a'];
        }
        else if (c == 164 || c == 165) ++nletras[26]; /* para la ñ */
    printf("Frecuencia de ocurrencia de letras \n");
    for (i = 'a'; i <= 'z' ; ++i) printf(" %c = %d\n", i, nletras[i-'a']);
    printf(" %c = %d\n", 164, nletras[26]);
    return 0;
}
```

Ejemplo2: uso de strcmp

```
/* **** */
* Programa: ejem_strcmp.c *
* Descripción: Prog. que usa funciones de comparacion y copia de strings *
\ **** /
#include <string.h>
#include <stdio.h>

void main( void )
{
    char tmp[20]; int result;
    char string1[] = "El perro marron rapido salta sobre el zorro loco";
    char string2[] = "El PERRO marron rapido salta sobre el zorro loco ";

    /* Case sensitive */
    printf( "Compara cadenas:\n\t%s\n\t%s\n\n", string1, string2 );
    result = strcmp( string1, string2 );
    if( result > 0 )
        strcpy( tmp, "mayor que" );
    else if( result < 0 )
        strcpy( tmp, "menor que" );
    else
        strcpy( tmp, "igual a" );
    printf( "\tstrcmp: String 1 es %s string 2\n", tmp );
}
```
