

Array de estructura sin conocer cant. de elementos

Escrito por Daniel1952 - 12/04/2015 15:12

Hola gente.

El programa que estoy construyendo tiene una búsqueda por fecha pero desconozco cuantos movimientos van a coincidir con la fecha a buscar.-

Si esto tiene que ver con memoria dinámica no lo puedo seguir porque si bien me falta muy poco para finalizar con el capítulo archivos memoria dinámica es el siguiente.-

Desconozco si se puede crear la estructura y luego en cualquier función declarar un array del tipo struct, es todo, no pongo el programa porque tiene más de 200 líneas y considero que no se necesita para evacuar mi consulta, por las dudas aclaro que la consulta se obtiene desde un archivo.-

Lenguaje C.

Desde ya muchas gracias por el tiempo que le dediquen.-

Saludos.

Daniel

=====

Re: Array de estructura sin conocer cant. de elementos

Escrito por adriánvaca - 14/04/2015 18:35

La forma de hacer eso es justamente con memoria dinámica, en la faq del sitio tienes ejemplos.

La otra forma es declarar un arreglo grande, por ejemplo de 100 elementos y llevar un contador para saber cuántos de esos 100 elementos estás usando.

=====

Re: Array de estructura sin conocer cant. de elementos

Escrito por Daniel1952 - 14/04/2015 19:26

Hola Adrián.

Muchas gracias por la pronta respuesta y por ocuparte del tema, estoy de acuerdo, sólo dime si no deseo desperdiciar memoria y lo hago de esta manera, que problema puedo tener.-

Código: #include

```
typedef struct{
    int va;
    float vb;
}Temp;
```

```
void fa( void );
```

```
int main(void){
```

```
fa();

return 0;
}

void fa( void ){
int vfa = 10;
Temp temp;

temp.va = 10;
temp.vb = 12.59;
printf(" n "va" vale -> %d == "vb" vale -> %.2f", temp.va, temp.vb); getch();
}código
```

Saludos.
Daniel

=====

Re: Array de estructura sin conocer cant. de elementos

Escrito por adriánvaca - 23/04/2015 17:11

El código está perfecto, la idea es no gastar memoria, al menos no mucha.

Con 10 elementos no habría problemas, si manejas 100, 1000 o incluso más sería casi obligatorio el uso de memoria dinámica.

=====