Programación 2009

Tarea 3

Las preguntas precedidas por un asterísco son para los alumnos de maestría. En licenciatura, dan puntos extras.

1 Tipos compuestos

Ejercicio 1 *Consideremos el tipo "double" (flotantes sobre 64 bits).*

1. *¿cuál sería el tipo de alineamiento a seguir en un sistema **32 bits**? ¿ 4, 8 ó 16 octetos ?*
2. *¿cuál sería el tipo de alineamiento a seguir en un sistema **64 bits**? ¿ 4, 8 ó 16 octetos ?*

Ejercicio 2 *Sea la siguiente unión:*

```
union u {
 short s;
 unsigned int i;
 char c;
} toto;
```

¿Hay un problema de alineamiento de datos con esa unión en una máquina 32 bits ? Explicar.

***Ejercicio 3** 1. *Escribir una estructura que permita representar puntos del espacio (en coordenadas x, y, z flotantes doble precisión). Luego, escribir una función "equal" que tome de entrada dos de esas estructuras y regrese 1 si los puntos son iguales, 0 si no.*

2. *Escribir una estructura que permita representar paralelepípedos con dos puntos extremos (x,y,z mínimos y x,y,x máximos), usando la estructura definida en el 1. De la misma manera, escribir una función de igualdad y una función de validez (que verificará que uno de los puntos corresponde a coordenadas mínimas, otra a coordenadas máximas).*

2 Apuntadores y alocación dinámica

Ejercicio 4 *Escribir un programa que haga la alocación de memoria dinámica para un arreglo de enteros unsigned int de tres dimensiones* $(100, 100, 100)$*, con triple apuntador. Rellenar el arreglo construido con valores (pseudo-)aleatorias y liberar finalmente este arreglo.*

Ejercicio 5 *Explicar por qué el programa siguiente no está bien (aunque compila) y proponerle un cambio para que sea funcional.*

```c
#include <stdlib.h>
void reinit(int *ptr) {
        if (ptr!=NULL) free(ptr);
        ptr = (int *)calloc(10,sizeof(int));
}
int main() {
  int *a = (int *)calloc(10,sizeof(int));
  reinit(a);
}
```

***Ejercicio 6** *Existe una función realloc que intenta cambiar el tamaño de un bloque de memoria previamente alocado con malloc o calloc.*

```
 void *realloc(void *ptr, size_t size);
The realloc() function shall change the size of the memory object pointed to by ptr
to the size specified by size. The contents of the object shall remain unchanged up
to the lesser of the new and old sizes. If the new size of the memory object would
require movement of the object, the space for the previous instantiation of the object
 is freed. If the new size is larger, the contents of the newly allocated portion of the
 object are unspecified. If size is 0 and ptr is not a null pointer, the object pointed
  to is freed. If the space cannot be allocated, the object shall remain unchanged.
```

¿Qué falta para que podamos implementar una función ***myrealloc*** *nuestra, con el mismo prototipo y con el mismo comportamiento que realloc ? Modificar ligeramente el prototipo de la función para poder hacerlo e implementar este* ***myrealloc****.*