

TRABAJO PRÁCTICO N°8

Deben entregarse por grupos en la entrega correspondiente vía Campus los archivos .c correspondientes a los ejercicios indicados.

1. Indicar el tipo de persistencia (automática o estática), el alcance (bloque, función o archivo) y el tipo de enlace (interno o externo) de todas las variables.

```
#include <stdio.h>

static int x; //Ej: estática, global y enlaceinterno
int a, b;

int main(void)
{
    int k;
    fun1();

    return 0;
}

void fun1(void)
{
    static int last;
    int h;
}
```

```
#include <stdio.h>

extern int x;
static int fun2(void);

int main(void)
{
    fun2();
    return 0;
}
```

2. Dado un programa en C compuesto de varios archivos separados, cuyo árbol de inclusión es el siguiente:

Archivo	Headers incluidos
main.c	stdio.h, process1.h, input.h, output.h
input.c	stdio.h, input.h, process2.h, list.h
output.c	stdio.h, output.h
process1.c	stdio.h, process1.h
process2.c	stdio.h, process2.h, list.h

- ¿Qué archivos deben ser recompilados luego de hacer cambios en process1.c?
 - ¿Qué archivos deben ser recompilados luego de hacer cambios en process1.h?
 - Escribir un makefile para el programa.
3. Escribir una función que imprima el número de veces que fue invocada.
4. ¿Por qué el siguiente programa no compila? ¿Cómo lo arreglaría, dejando la variable x en el archivo file2.c?

```
file1.c

#include <stdio.h>

int main()
{
    printf("%d\n", x);
    return 0;
}
```

```
file2.c

static int x = 5;
```

5. **[ENTREGAR]** Reescribir el ejercicio 14 del TP 4 (Juego de la vida) en formato multiarchivos (*También corregir los errores que se hayan indicado en la corrección del TP, si los hubiere*). Escribir también un archivo makefile para su compilación (que aproveche todas las ventajas de make). Prueben compilar de dos maneras distintas: con make mediante el makefile; y llamando directamente a gcc desde la línea de comandos. **Incluyan en la entrega el makefile que escribieron y un archivo "compilacion.txt" que contenga el comando completo que usaron para compilar por línea de comandos.**

Recuerden enfatizar los siguientes puntos:

- Modularidad
- Diferenciación entre funciones particulares y funciones reutilizables o de propósito general
- COMENTARIOS: Recuerden los métodos vistos en clase y aplíquenlos, especialmente en los archivos .h. Si le entregan únicamente este archivo a otra persona, ¿será capaz de utilizarlo sin problemas? Ustedes mismos, ¿se ven entendiendo el funcionamiento en el futuro?