TP N°1: Compilación y Linkedición

La siguiente guía cubre los contenidos vistos en las clases teóricas:
1. Introducción a la materia
2. Introducción al lenguaje C

Antes de comenzar con la guía, se recomienda la lectura del apunte **Tutorial de Compilación en Linux** para comprender el uso del compilador gcc. Disponible en Campus ITBA.

Se recomienda para todo el transcurso de la materia **utilizar los siguientes flags de gcc**.
-Wall Lista más advertencias del código
-pedantic Realiza las validaciones con el estándar de C
-std=c99 Para forzar el estándar ANSI C ISO C-99 recomendado por la cátedra

Ejercicio 1

Escribir y compilar el siguiente programa. Anotar los errores de compilación encontrados.

```
int
main(void)
{
      /* Esto es un primer comentario
      */
      \* Que buen comentario! *\
      /* Es valido /* o no */  este comentario?  */
      /      *  Esto es un ultimo comentario */

return 0
}
```

Ejercicio 2

Idem al ejercicio 1.

```
int
main( void /* programa que no hace nada */)
{
      return 0
}
```

Ejercicio 3

Idem al ejercicio 1. Corregir los errores y correr el programa.

```
#include <stdio.h>

int
Main(void)
{
      printf ("Estamos escribiendo un mensaje \n")
      Return = 0
}
```

Ejercicio 4

Idem al ejercicio 3.

```
#include <stdioh>

int
main[void]
{
      int a, b = c = 5
      a = b + c

      return: 0
}
```

Ejercicio 5

Idem al ejercicio 3.

```
#      include <stdio.h>

int
main(void)
{
      int i, j, max;
      i = j = 2;
      max = (i>j? I : j ) ;

      end
}
```

Ejercicio 6

Compilar pero no linkeditar el archivo **tp1_06.c** ¿Con qué atributos (permisos) se crea el archivo resultante? Intentar ejecutarlo.

Recordá que si sólo se desea compilar (obtener el código objeto), es necesario utilizar el flag -c. Se recomienda utilizar esta opción para diferenciar claramente los errores de compilación y linkedición.

$> gcc -c tp1ej6.c -Wall -pedantic -std=c99

Ejercicio 7

Compilar y linkeditar (en pasos separados) el archivo **tp1_07.c**. Indicar en qué paso falla y por qué.

Ejercicio 8

Idem ejercicio anterior con archivo **tp1_08.c**

Ejercicio 9

Compilar los archivos **tp1_09a.c** y **tp1_09b.c,** y linkeditarlos juntos. ¿En qué paso falla la traducción y por qué?

Para compilar y linkeditar juntos a dos archivos fuentes los vamos a mandar a ambos como parámetros de entrada del gcc así:

$> gcc tp1_09a.c tp1_09b.c -o tp1_09 -Wall -pedantic -std=c99

Ejercicio 10

Compilar y linkeditar el archivo **tp1_10.c** con la opción **-o tp1_10** y probar los siguientes comandos:

- **tp1_10**
- **tp1_10 | cat**
- **tp1_10 > out.txt**

Explicar el por qué de cada salida.

Ejercicio 11

Compilar el programa **tp1_11.c** . Si se produce un error corregirlo y luego probar los siguientes comandos:

- **tp1_11 < tp1_11.txt**
- **tp1_11 < tp1_11.txt > out.txt**

Explicar el por qué de cada salida.

Ejercicio 12

Preprocesar el programa **tp1_12.c** y analizar la salida.

> Con la opción -E, gcc se detiene luego del preprocesamiento.
> $> gcc -E tp1ej12.c

Ejercicio 13

Compilar, linkeditar y ejecutar el programa **tp1_13.c**. Hacer lo mismo con el programa **tp1_13b.c**. Analizar las diferencias entre ambos.

Ejercicio 14

Compilar y linkeditar el programa **tp1_14.c**. Hacer lo mismo con el programa **tp1_14b.c**. Analizar diferencias con el ejercicio anterior.

Ejercicio 15

¿Por qué el siguiente código falla en la compilación?

```
#include <stdio.h>

int
main(void)
{
        int manual = 7;
        int auto = 9;
        /* Se calcula el promedio */
        return 0;
}
```

> Revisá las primeras diapositivas de la clase teórica 2. Introducción al lenguaje C

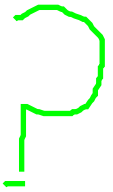
Ejercicio 16

Indicar, para el escenario listado, si gcc falla o no en la construcción de un programa ejecutable. Si falla, decir si lo hace en la etapa de preprocesamiento, compilación o linkedición:

1. Archivo de encabezado .h inexistente
2. Más de una función main() en un único archivo fuente .c
3. Más de una función main() en un conjunto de archivos fuente .c
4. Ninguna función main() presente en un conjunto de archivos fuente .c
5. Uso de una palabra reservada del lenguaje como nombre de una variable
6. Redefinición de una variable dentro de un bloque o función (en el siguiente ejemplo, la variable x está definida dos veces):

```
#include <stdio.h>

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



7. La función main() no tiene la línea return 0; o similar al final del cuerpo de la misma
8. Dos funciones con el mismo nombre en el mismo archivo fuente .c

Cuando veas que gcc lista un error con el término ld, significa que falló la linkedición. Si el error es de linkedición entonces el preprocesamiento y la compilación de todos los archivos fuentes involucrados fue exitosa.