Prueba del proceso de compilacion por etapas

1. Escribir un programa hola.c: podrán observar que tiene errores, escríbanlo asi

```
#include <stdio.h>
#define CARGA_MAXIMA 1500
int/*medio*/main(void){
int i=42;
prontf("La respuesta es %d\n");
prontf("el valor %d\n",CARGA_MAXIMA);
```

Observen los errores o cosas “raras”

```
int/*medio*/main(void){
```

pueden ver un comentario en un lugar no habitual
se hace referencia a una función prontf, esta no esta declarada, quizá se pueda referie a printf declarada en la biblioteca, pero no esta bien escrita

Procedan ahora aPreprocesar hola.c , no compilar, y generar hola.i .
Y analizar su contenido.
Para ello deben, desde la línea de comando ejecutar

gcc -E hello2.c -o hello2.i PREPROCESO (.i)

vean que el resultado El " hola.i" contiene los structs, typedef, prototipos de las funciones y directivas del archivo cabecera "stdio.h" . Al final de " hola.i", el codigo del archivo " hola.c" escrito sin los comentarios
La directiva #include debe ir seguida del nombre de un fichero y su efecto es dereemplazar esta línea en el código fuente por el contenido del fichero que seespecifica.

#define CARGA_MAXIMA 1500
el preprocesador reemplaza toda aparición de la primera cadena por la segunda.

2. int printf(const char *s, ...);
int main(void){
int i=42;
prontf("La respuesta es %d\n");

aahora compilar no ensamblar, ejecutar:

gcc -S hola.c COMPILAR SIN ENSAMBLAR (.s)

Produce una ==advertencia== porque no esta declarada la función prontf.
Señala un ==error== .Elmismo indica que hay un error de sintaxis (falta un simbolo '}'). Crea el archivo hola.s pero sin contenido.

3. Poner la llave y ejecutar
gcc -S hola.c

Produce una ==advertencia== porque no esta declarada prontf. Esta vez, generó el archivo hola.s con codigo assembler.

Define la organización de un programa en assembler. Tiene seccion donde declara los datos y otra donde declara los pasos a ejecutar en assembler.

ANALISIS LEX->SINTACTICO CON EL ARBOL SINTACTICO Y SEMANTICO, SE OPTIMIZA Y SE GENERA ENSAMBLADOR

4. Ensamblar hola.s en hola.o , no vincularas hola.s -o hola.o ENSAMBLAR SIN VINCULAR (.o)

Se crea un archivo binario llamado hola.o que contiene código objeto.

5. Vincular hola.o con la biblioteca estándar y generar el ejecutable
gcc hola.o –o ejecutable vincular y generar el ejecutableresultado

hola.o: En la función `main':
hola.c:(.text+0x1a): referencia a `prontf' sin definir
collect2: error: ld returned 1 exit status
No encuentra la definición de prontf. No se crea el ejecutable.
Enlace con las bibliotecas disponibles en código máquina para generar el ejecutable.

6. generar el ejecutable.
comando ejecutado
gcc hola.c -o hola
Ejecutar

Como ven si bien cuando compilamos apretaban solo “un botoncito” los procesos son multiples y aquí se analizan por partes