

**PRACTICA: Creación de Aplicaciones en C (II)****Objetivos:**

- Manejo del Lenguaje C
- Uso de funciones de librería y ficheros cabecera
- Uso de compilador: gcc
- Crear librerías dinámicas y estáticas
- Gestionar los proyectos con make

1.- Dado los siguientes módulos en C (Ejercicio de examen)

**En /home/lsi/include/ tenemos los siguientes .h:****aritmetica.h:**

```
#ifndef _ARITMETICA_H
#define _ARITMETICA_H
int producto (int a, int b);
int resta (int a, int b);
int suma (int a, int b);
#endif
```

**constante.h**

```
#ifndef _CONSTANTE_H
#define _CONSTANTE_H
#define PI 3.1415
#endif
```

**area.h**

```
#ifndef _AREA_H
#define _AREA_H
int rectangulo (int x, int y);
float circulo (int x);
#endif
```

**En /home/lsi/fuentes/ tenemos****aritmetica.c:**

```
int suma (int a, int b) { return a+b; }
int resta (int a, int b) { return a-b; }
int producto (int a, int b) {return a*b; }
```

**area.c:**

```
int rectangulo (int x,int y){return producto(x,y);}
float circulo (int x){return (PI * producto(x,x));}
```

a) Incluye todos los `#include` en los ficheros “aritmética.c” y “area.c”

Aritmética.c:

```
int suma (int a, int b) { return a+b; }
int resta (int a, int b) { return a-b; }
int producto (int a, int b) {return a*b; }
```

Area.c:

```
#include <constante.h>
#include <aritmética.h>

int rectangulo (int x,int y){return producto(x,y);}
float circulo (int x){return (PI * producto(x,x));}
```

b) Realizar una aplicación principal en “/home/lsi/fuentes/examen.c”, que pida 2 números enteros por teclado y nos muestre por pantalla la suma de los dos números y el área del rectángulo, utilizando los módulos de arriba.

```
#include <stdio.h>
#include "aritmética.h"
#include "area.h"

int main()
{
    //DEFINICION
    int a,b;
    //ENTRADA VALORES
    printf("\nIntroduce un numero:");
    scanf("%i",&a);
    printf("\nIntroduce otro numero:");
    scanf("%i",&b);
    //RESULTADOS
    printf("\n\n\tLa suma de estos numeros es: %i",suma(a,b));
    printf("\n\tEl area del rectangulo formado por ambos numeros es: %i",rectangulo(a,b));
    //END
    return 0;
}
```

c) Compila cada módulo por separado y finalmente crea el ejecutable utilizando dichos módulos objeto (suponiendo que estos módulos no están en ninguna librería).

```
gcc -o examen.o -c examen.c -I../include/
gcc -o área.o -c area.c -I../include/
gcc -o aritmética.o -c aritmética.c -I../include/
gcc examen.o área.o aritmética.o -o examen -I../include/
```

d) Introduce área y aritmética en una librería estática que se llamará “libestatica.a” en un nuevo directorio /home/lsi/librerias/ y vuelve a crear la aplicación.

```
ar rv libestatica.a area.o aritmética.o
ranlib libestatica.a
gcc -L../librerias/ -l. -o examen examen.c -lestatica -static
```

e)Introduce área y aritmética en una librería dinámica que se llamará “libdinamica.so” en el nuevo directorio **/home/lsi/librerias/** y vuelve a crear la aplicación.

```
ld -o libdinamica.so area.o aritmetica.o -shared
```

```
gcc -o examen examen.c -I. -L../librerias/ -ldinamica -Bdynamic
```

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/xabierland/Documents/IOS/LAB4/librerias/
```

```
Export LD_LIBRARY_PATH
```

f)Realiza el apartado c) pero usando el comando make.

make

```
examen : area.c aritmetica.c examen.c
```

```
gcc -o examen area.c aritmetica.c examen.c -I../include/
```