Tarea Biblioteca Estática libIntroProgUtil.a

Desarrollo

Considere los siguientes archivos del proyecto ProEst02.cbp



En esta disposición de los archivos, las funciones

void copy\_matriz(struct matriz \*to,struct matriz \*from)

float det(struct matriz \*M)

float \*\*get\_traspose(float \*\*A,unsigned int rows,unsigned int cols)

struct matriz \*inv(struct matriz \*M)

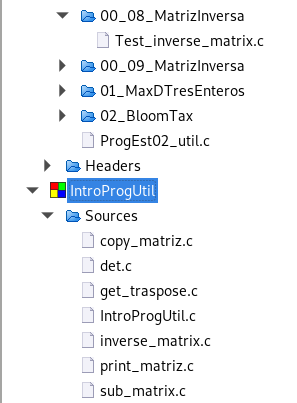
void print\_matriz(struct matriz \*M)

float \*\*MSUBM(struct matriz \*M, int row,int col)

void MultByScalar(float factor,float \*\*A,int r,int c)

struct matriz \*Mult(struct matriz \*A,struct matriz \*B)

solo se pueden utilizar dentro del proyecto ProgEst02. Para que estas funciones se puedan utilizar desde cualquier proyecto dentro de nuestro espacio de trabajo Workspace\_IntroProg\_2022\_2, mueva los archivos hacia el proyecto de biblioteca estática IntroProgUtil, como se muestra en la siguiente figura:



Para que la biblioteca estática IntroProgUtil se pueda construir sin errores, usted deberá ajustar las rutas relativas de ubicación de los archivos de cabecera, como por ejemplo, en el archivo print\_matriz.c, las líneas de código relevantes pueden quedar como sigue:

/\*\* print\_matriz.c \*/

#include <stdio.h>

#define CONFIG\_USING\_LIB\_INTROPROGUTIL

#ifndef CONFIG\_USING\_LIB\_INTROPROGUTIL

#include "struct\_matriz.h"

#else

#include "../202110/ProgEst02/include/struct\_matriz.h"

#endif // CONFIG\_USING\_LIB\_INTROPROGUTIL

/\*\*

\* Imprime la matriz a la que apunta el

\* apuntador M.

\*/

void print\_matriz(struct matriz \*M)

// …

Nota: en este ejemplo, las directivas están colocadas con la finalidad de hacer más facil intercambiar este archivo de un proyecto a otro. Por ejemplo, para regresar el archivo print\_matriz.c al proyecto ProgEst02.cbp, después de agregar el archivo print\_matriz.c a este último proyecto, simplemente se tendría que comentar la línea que contiene #define CONFIG\_USING\_LIB\_INTROPROGUTIL, para que quede así:

//#define CONFIG\_USING\_LIB\_INTROPROGUTIL

Sin embargo, si el uso de las directivas le resulta confuso, en una primera aproximación puede usted omitirlas y simplemente cambiar la línea

#include "struct\_matriz.h"

por la línea

#include "../202110/ProgEst02/include/struct\_matriz.h"

y con ello el archivo print\_matriz.c deberá poder compilarse sin errores. Por otra parte para corregir los warnings que se nos presentarán por el uso de la función void print\_matriz(struct matriz \*M), como el siguiente:

warning: implicit declaration of function ‘print\_matriz’ [-Wimplicit-function-declaration]

simplemente deberemos colocar el prototipo de la función

void print\_matriz(struct matriz \*M)

antes de la utilización de la misma; por ejemplo, podemos colocar este prototipo en alguno de los archivos de cabecera que incluiremos en el archivo Test\_inverse\_matrix.c, como se muestra a continuación:

/\*\* Test\_inverse\_matrix.c \*/

#include <stdio.h>

#include <stdlib.h>

#include "struct\_matriz.h"

#include "../include/ProgEst02\_util.h"

int main(int argc,char \*argv[])

// …

Podemos elegir colocar el prototipo por ejemplo en el archivo ../include/ProgEst02\_util.h:

/\*\* ProgEst02\_util.h \*/

// ...

/\*\*

\* Imprime la matriz a la que apunta el

\* apuntador M.

\*/

void print\_matriz(struct matriz \*M);

#endif // PROGEST02\_UTIL\_H\_INCLUDED

De manera similar, deberá usted corregir los errores y warnings que se presentarán al cambiar los archivos copy\_matriz.c, det.c, get\_traspose.c, inverse\_matriz.c, print\_matriz.c, y sub\_matrix.c del proyecto ProgEst02.cbp al proyecto IntroProgUtil.cbp.