



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

E. A. Nro. 2– Segunda Evaluación de Aprendizaje.

Debe resolver la siguiente problemática eligiendo la mejor estrategia.

Parte 1,-.POO.- Un muy simple y muy conceptual ejercicio de Programación Orientada a Objetos en Lenguaje C++. Debe completar la declaración de la clase Prisma (en **Prisma.h**). Una vez hecho debe asegurarse que el proyecto compila sin errores (entrando por el menú desplegable [**Build**] / [**Compile current file**]). Luego, complete la definición de la clase (en **Prisma.c**). Una vez hecho, compile y ejecute. En la carpeta del proyecto verá que hay un programa ejecutable (**EA-2-Prisma.exe**) y un archivo de comando (**EJECUTAR.cmd**). Haciendo "doble click" en el archivo de comando, verá la ejecución esperable del programa que usted debe resolver (además de generarla en un archivo de texto). No debe modificar ni **main.c** ni **main.h**. Debe enviar los archivos "**Prisma.c**" y "**Prisma.h**" del modo indicado al final de este documento. La evaluación consiste en que la salida que se produce sea la que se muestra y se espera. Además, se verá la correctísima declaración y definición de la clase, sus métodos y funciones miembro.

Una nota aclaratoria: será preferible que el atributo de información de la clase Prisma para el color sea **string**, de lo contrario **char ***, tal como se indica en **Prisma.h**.

NOTE que el proyecto entregado no podrá compilar ni generar el ejecutable hasta que lo haya terminado.

Parte 2.-.TDA.- Se dispone de una función que permite cargar información sintética (simulada) de los movimientos de las cuentas de los clientes de un banco. Esta información está compuesta de: identificador de cuenta bancaria e importe ⁽¹⁾, que se supone están en orden cronológico en la lista. Para consolidar el saldo de los clientes se requiere en primer lugar ordenar la lista ⁽²⁾, por el identificador de cuenta bancaria en



modo ascendente, de modo que para cada cuenta si hay más de un movimiento que en éstos se respete el orden cronológico.

Una vez ordenada, se necesita eliminar de la misma los movimientos de cuentas corrientes (cuentas que terminan en 002 ⁽¹⁾). Se requiere que los que corresponden al mismo cliente (o identificador de cuenta bancaria) se '*muestren*' en el mismo orden relativo en que quedaron después del ordenamiento y que al cambiar de cliente se muestre el importe total de los movimientos del cliente. La salida de su programa debe coincidir exactamente con la que hace el proyecto que se le entrega. No debe alterar los archivos `funciones.h` ni `main.h`. Las únicas modificaciones en `main.c` son para que invoque sus primitivas.

⁽¹⁾ El identificador de cuenta bancaria está compuesto por tres grupos de dígitos separados por '-' (guion). Los primeros tres dígitos identifican la sucursal. Los siguientes siete identifican al cliente de la sucursal (en otra sucursal otra persona física o jurídica puede tener los mismos siete dígitos). Los últimos tres identifican el tipo de cuenta (001 caja de ahorro, 002 cuenta corriente en pesos, 003 otro tipo de cuenta -podrá haber más tipos de cuenta-).

- Dispone de un proyecto que hace todo lo pedido, con lo que puede comparar los resultados que obtiene con los resultados esperados.

Debe resolver las primitivas de mostrar la lista y vaciarla como se vieron en los apuntes de clase.

Además, la primitiva que permite ordenarla ^(*), empleando el método de ordenamiento que se detalla más abajo y la primitiva que permite eliminar los movimientos de cuentas corrientes mostrando el total de cada cliente (como se ve en el proyecto).

^(*) Para el ordenamiento de la lista debe hacer lo siguiente, por cada uno de los nodos de la lista: buscar el menor de los menores entre todos los que están a su derecha (que



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

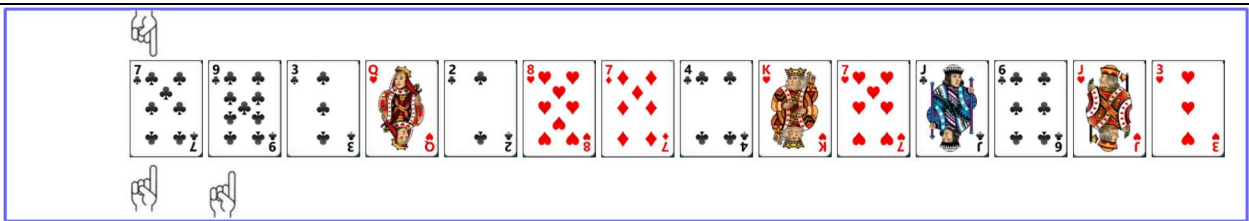
también sea menor que él) quitarlo de esa posición e insertarlo inmediatamente antes del nodo. A continuación se ilustra la estrategia **que debe implementar** para el ordenamiento.

Dado un "desorden" cualquiera, por ejemplo, en un grupo de cartas del descarte, si inicialmente éstas resultaran dispuestas así (ver la siguiente figura):

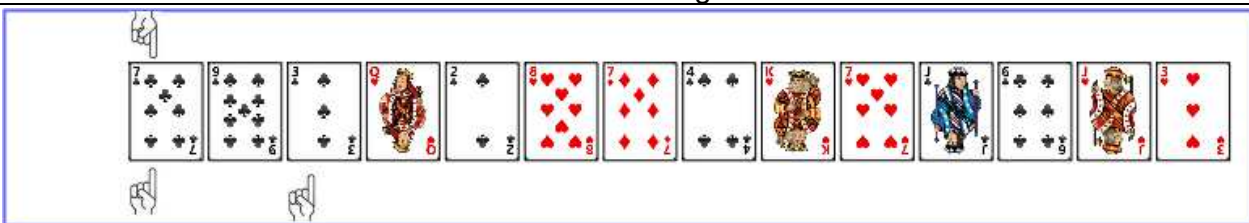
7T – 9T – 3T – QC – 2T – 8C – 7D – 4T – KC – 7C – JT – 6T – JC – 3C

Si hay que ordenar por el número (no por color), respetando a igualdad de número el orden relativo en que están originalmente. El resultado final será:

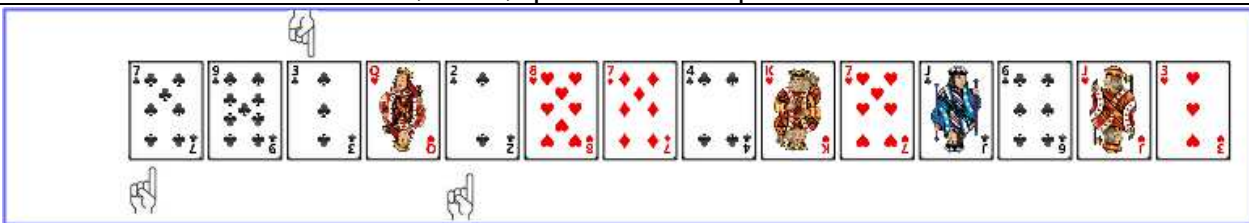
2T – 3T – 3C – 4T – 6T – 7T – 7D – 7C – 8C – 9T – JT – JC – QC – KC



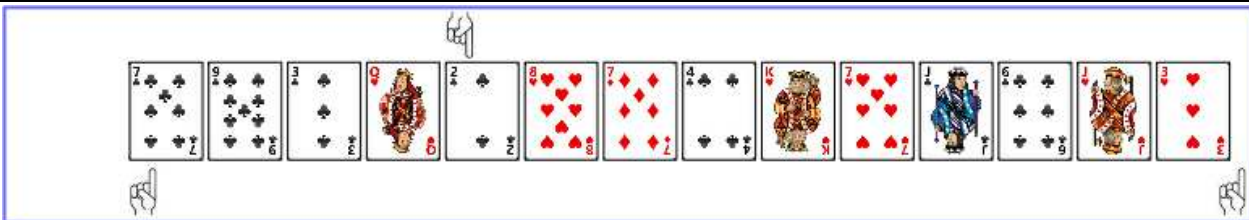
Para el ordenamiento se comienza como en la figura anterior con el 7T.



Cuando encuentra una carta, el 3T, que es menor que el 7T . . .



Desde ese momento el 3T es la menor. Continúa buscando y encuentra el 2T que es menor que el 3T . . .

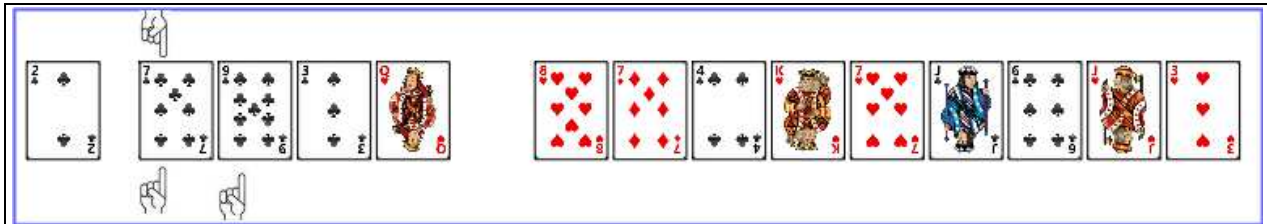


. . . con lo que el 2T pasa a ser la menor carta hasta el momento. Continúa buscando sin encontrar otra que sea menor . . .

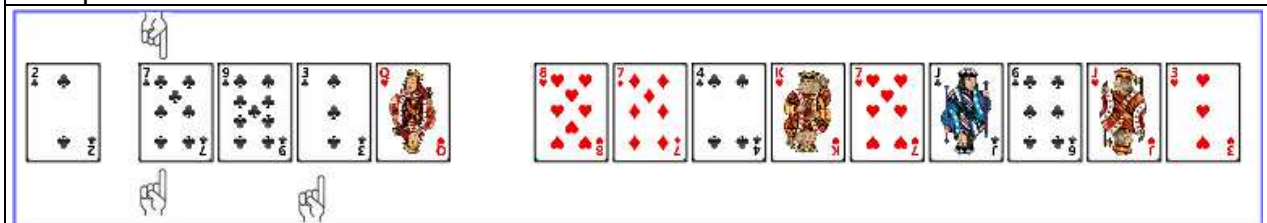


UNLaM

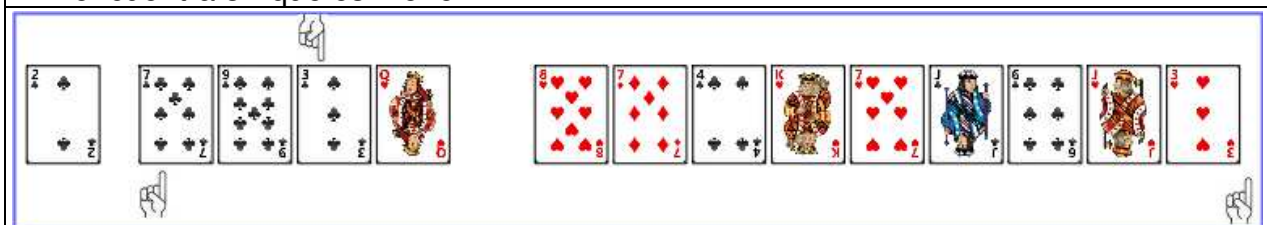
Dto. Ingeniería e Investigaciones Tecnológicas



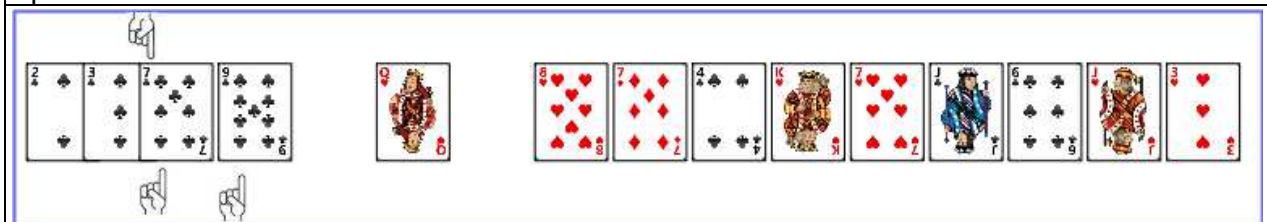
... por eso se saca el 2T y se la pone antes del 7T. Se prepara para repetir la búsqueda de la menor ...



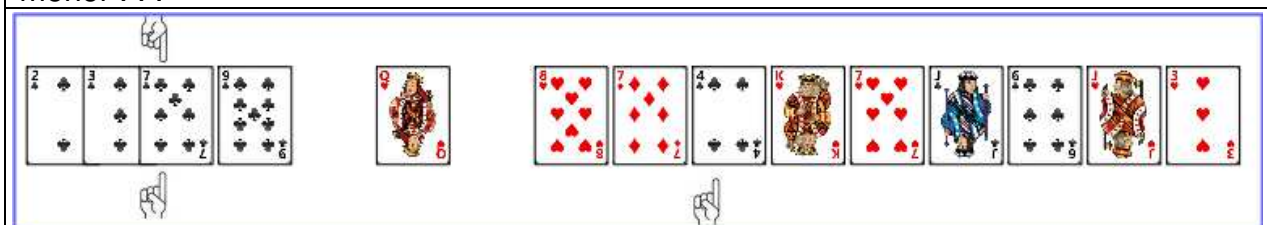
... encuentra 3T que es menor ...



... 3T pasa a ser la menor hasta el momento. Continúa buscando sin encontrar otra que sea menor ...



... por eso el 3T se pone antes del 7T. Se prepara para repetir la búsqueda de la menor ...



... encuentra 4T que es menor que 7T ...



UNLaM

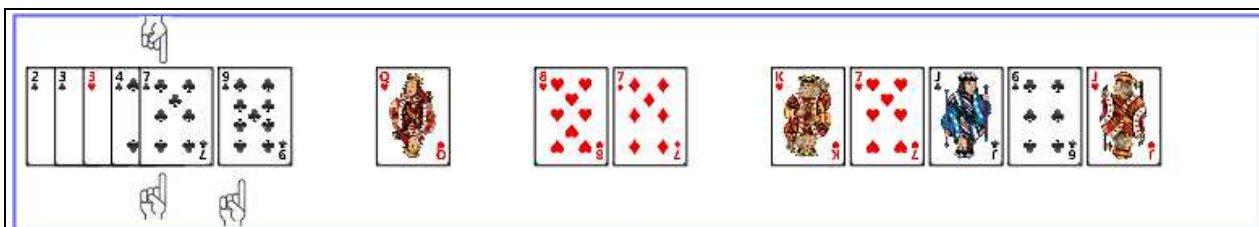
Dto. Ingeniería e Investigaciones Tecnológicas

...	4T	pasa a ser la menor hasta el momento. Continúa buscando y encuentra 3C menor que 4T...
...	3C	pasa a ser la menor hasta el momento. Continúa buscando y no hay más...
...	3C	se saca de dónde estaba y se la pone antes de 7T. Se prepara para repetir la búsqueda de la menor...
...	4T	encuentra 4T que es menor que 7T...
...	4T	pasa a ser la menor hasta el momento. Continúa buscando y no encuentra más...

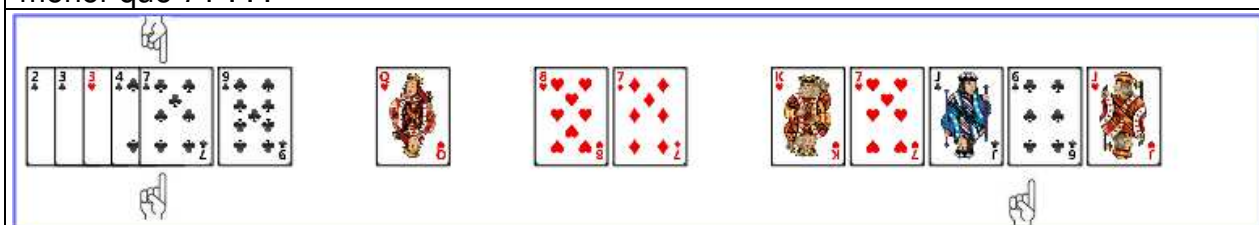


UNLaM

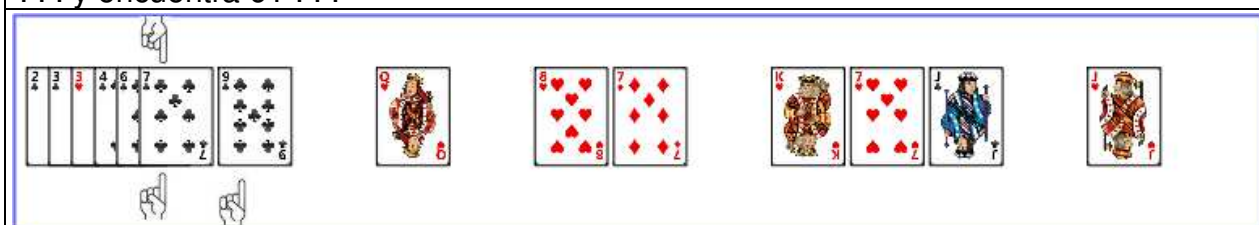
Dto. Ingeniería e Investigaciones Tecnológicas



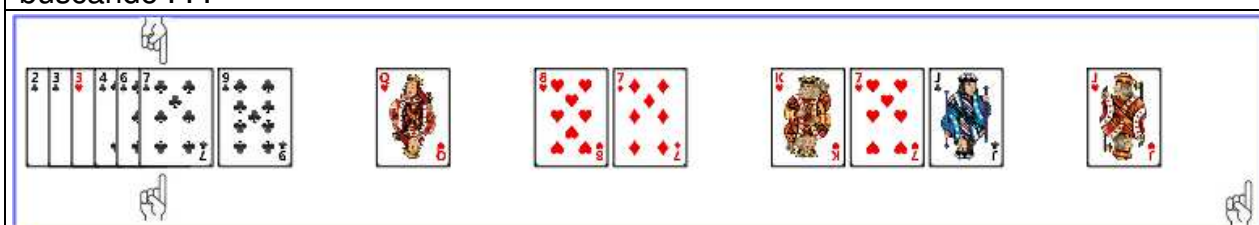
... por eso 4T se pone antes de 7T. Se prepara para repetir la búsqueda de una menor que 7T ...



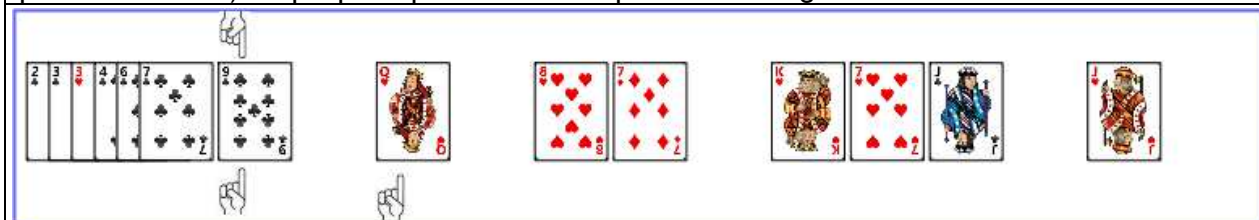
... y encuentra 6T ...



... 6T es la menor hasta ahora ... al seguir buscando una menor que 6T no hay otra que sea menor, se saca 6T y quedará antes que 7T. Y se prepara para seguir buscando ...



... como no encuentra ninguna menor que 7T, (todas las menores se sacaron y se pusieron antes) se prepara para buscar a partir de la siguiente a 7T ...



... que resulta ser el 9T (podía ser cualquier otra mayor o igual a 7T). Y partir de ahora repite el mismo procedimiento ...



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

. . . con lo que se repite la búsqueda de la menor de las menores de 9T, con eso pondrá antes del 9T el 7D, luego el 7C, luego el 8C.

Al no quedar menores que 9T, procede a buscar las menores que QC, poniendo antes JT y luego JC y al no haber más menores . . .



. . . pasará a buscar las menores que KC . . .



. . . y como es la última y no hay siguiente . . . **Terminó**



Acá, una vez extendidas las cartas, estas quedan ordenadas del modo esperado, por número pero respetando el orden en que estaban a igualdad de número.

2T – 3T – 3C – 4T – 6T – 7T – 7D – 7C – 8C – 9T – JP – JC – QC – KC

Debe repasar y aplicar esta estrategia, en la que se le ha brindado una explicación más que significativa del modo a emplear en la solución del algoritmo (primitiva de ordenamiento de una lista simplemente enlazada con asignación dinámica de memoria).

¡Se debe utilizar este método de ordenamiento y no otro!

Cualquier otra solución será inválida.



Consideraciones Generales

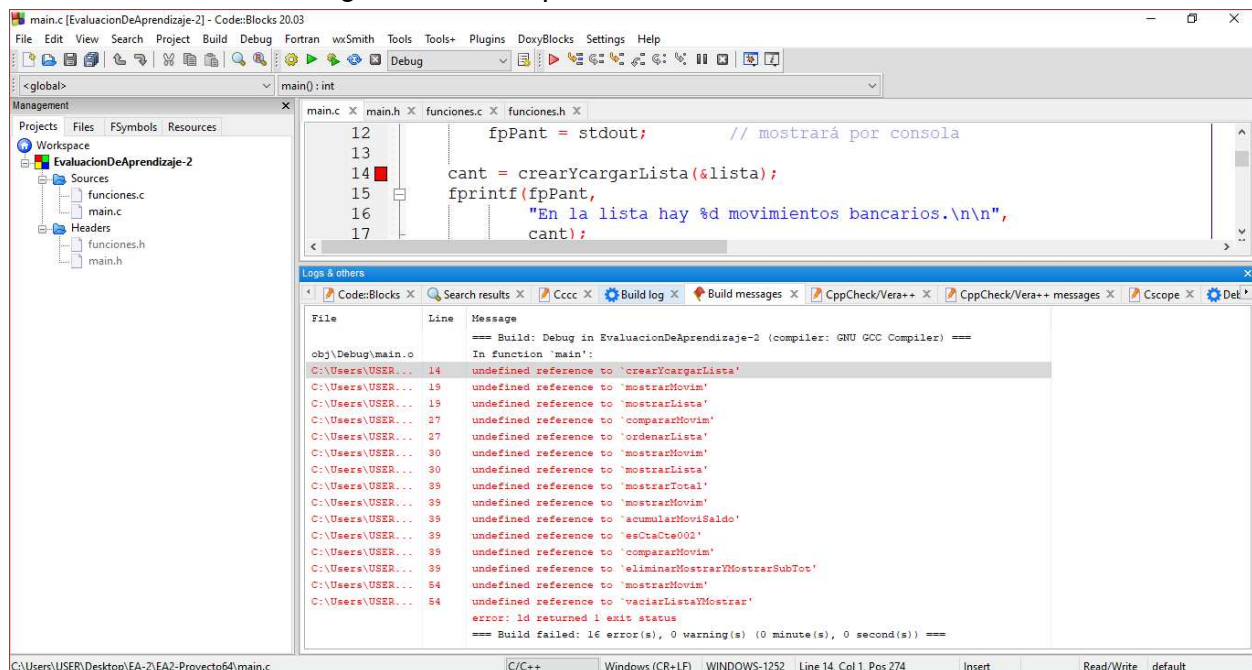
Descargue el proyecto provisto, genere una *carpeta* en su computadora y en ella descomprímalo.

Verá que se generan dos *subcarpetas*, una para plataforma (o compilador) de 32 bits y otro para plataforma de 64 bits.

En la carpeta que corresponde a su IDE (**EA2-Proyecto64** o **EA2-Proyecto32**) abra el proyecto que corresponde a su plataforma (habitualmente doble click en el archivo **EA-2.cbp** o en su defecto una vez abierto el IDE, "*arrastrándolo*" en el mismo).

Proceda a compilarlo.

Si en el cuadro de diálogo inferior le aparecen errores:



... ¡¡¡es porque se equivocó de proyecto!!! Vaya al otro y compile:



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

```

1  #include "main.h"
2
3
4  int main()
5  {
6      FILE *fpPant = fopen("pantalla.txt", "wt");
7      tLista lista;
8      int cant;
9      cantElim;
10
11      if(fpPant == NULL) // si no puede abrir el archivo
12          fpPant = stdout; // mostrará por consola
13
14      cant = crearYcargarLista(&lista);
  
```

```

----- Clean: Debug in EvaluacionDeAprendizaje-2 (compiler: GNU GCC Compiler)-----
Cleaned "EvaluacionDeAprendizaje-2 - Debug"

----- Build: Debug in EvaluacionDeAprendizaje-2 (compiler: GNU GCC Compiler)-----
gcc.exe -Wall -g -c C:\Users\USER\Desktop\EA-2\EA2-Proyecto64\funciones.c -o obj\Debug\funciones.o
gcc.exe -Wall -g -c C:\Users\USER\Desktop\EA-2\EA2-Proyecto64\main.c -o obj\Debug\main.o
gcc.exe -o bin\Debug\EA-2.exe obj\Debug\funciones.o obj\Debug\main.o .\lib-EA_2_Release_bib.a
Output file is bin\Debug\EA-2.exe with size 60.01 KB
Process terminated with status 0 (0 minute(s), 0 second(s))
0 error(s), 0 warning(s) (0 minute(s), 0 second(s))
  
```

Cuando compile el proyecto, asegúrese que el IDE genera el ejecutable con las opciones como se ve en la figura anterior.

gcc.exe -Wall -g -c C:(...etcétera)

gcc.exe -Wall -g -c C:(...etcétera)

gcc.exe -o bin\Debug\(...etcétera) .\lib-EA_2_Release_bib.a

De otro modo, podrá tener algunos problemas en los que no le podremos ayudar.

DEBE DEVOLVER TAN SOLO los archivos **funciones.c**, **Prisma.c** y **Prisma.h** y al comienzo de cada uno de estos archivos, debe poner en una línea de comentario con:

DNI-APELLIDO,Nombre-(curso-comisión), por ejemplo:

// 22.333.444-PEREZ_DEL_RIO, JuanManuel-(07-2299) *//**/**

Estos archivos deben estar comprimidos únicamente en formato **.zip (*)**. Para ello copie los tres archivos en una misma ubicación, selecciónelos y con botón derecho elija de la ventana emergente la opción: **[Enviar a] / [Carpeta comprimida (en zip)]** (o **[Send to] / [Compressed (zipped) folder]**)

EL archivo debe estar renombrado con su número de dni sin puntos, por ejemplo:

22333444.zip

En el asunto o motivo debe indicar tan sólo: **EA-2**