

```
main.c x main.h x fechas.c x fechas.h x
1  #ifndef FECHAS_H_
2  #define FECHAS_H_
3
4  #include <stdio.h>
5
6
7  /**
8   * tipo de dato para la fecha, los miembros de la 'struct' podrían estar en
9   * otro orden.
10  */
11  typedef struct
12  {
13      int di,
14          me,
15          an;
16  } tFecha;
17
18  /**
29  int ingresarFechaDMA(tFecha *fec);
30
31  /**
34  void mostrarFechaDMA(const tFecha *fec);
35
36
37
38
39
```

main.c x main.h x fechas.c x fechas.h x

```
1
2  #include "fechas.h"
3
4  int ingresarFechaDMA(tFecha *fec)
5  {
6      printf("Fecha (dd/mm/aaaa): ");
7      fflush(stdin);
8      scanf("%d/%d/%d", &fec->di, &fec->me, &fec->an);
9      return 1;
10 }
11
12 void mostrarFechaDMA(const tFecha *fec)
13 {
14     printf("%02d/%02d/%04d", fec->di, fec->me, fec->an);
15 }
16
17
18
19
20
21
22
23
24
25
26
27
```

```
1
2  #include "main.h"
3
4
5  int main()
6  {
7      tFecha  fec;
8
9      while(ingresarFechaDMA(&fec))
10     {
11         mostrarFechaDMA(&fec);
12         puts("");
13     }
14     puts("Quiso terminar");
15
16     return 0;
17 }
18
19
20
21
22
23
24
25
26
27
```

main.c x main.h x fechas.c x fechas.h x

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19

```
#ifndef MAIN_H_
#define MAIN_H_

#include <stdio.h>

#include "fechas.h"

#endif
```

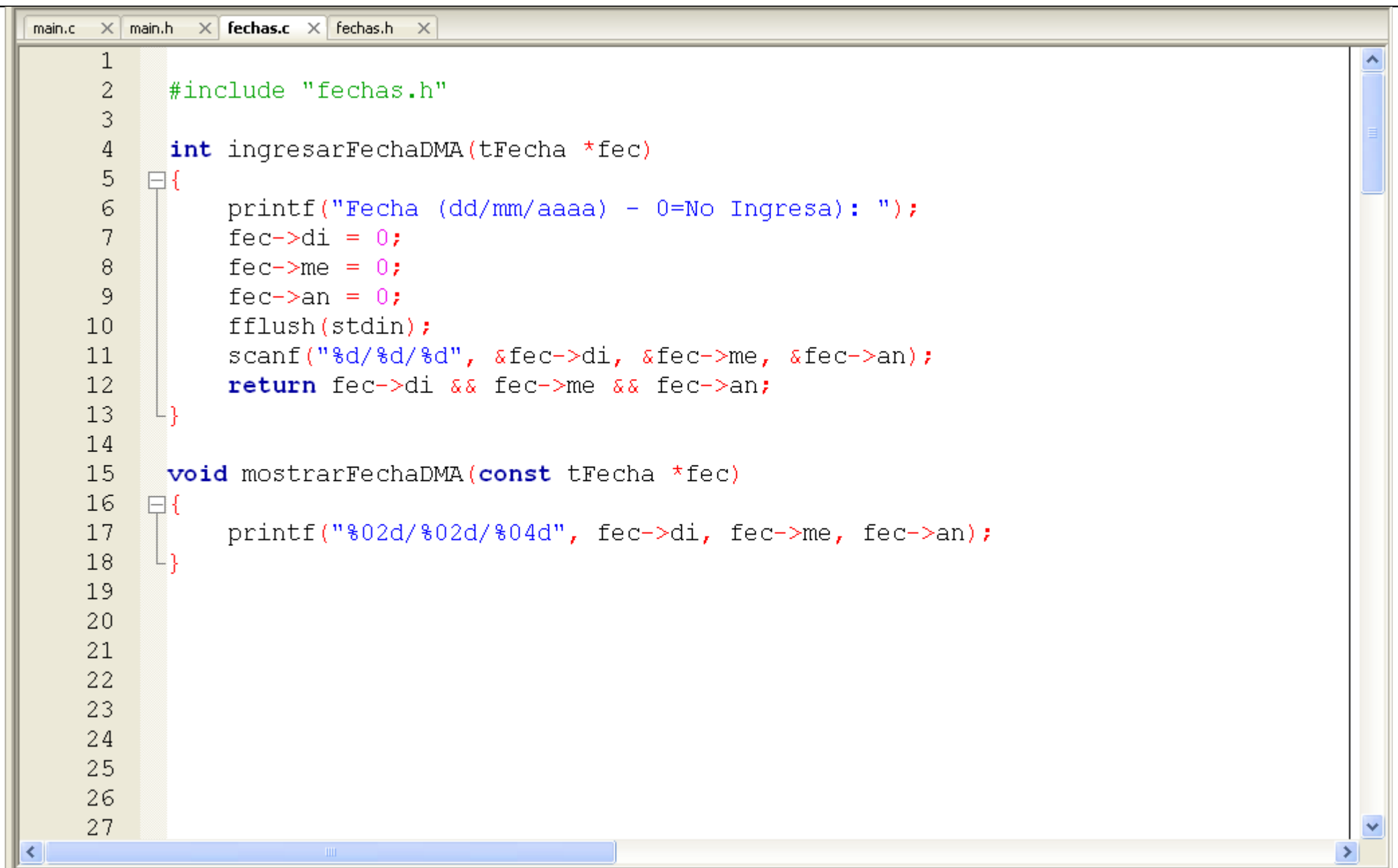
Logs & others x

Code::Blocks x Search results x Build log x Build messages x Debugger x

```
mingw32-gcc.exe -Wall -g -c C:\Proyectos-CodeBlocks\fechas\fechas.c -o obj\Debug\fechas.o
mingw32-gcc.exe -Wall -g -c C:\Proyectos-CodeBlocks\fechas\main.c -o obj\Debug\main.o
mingw32-g++.exe -o bin\Debug\fechas.exe obj\Debug\fechas.o obj\Debug\main.o
Output file is bin\Debug\fechas.exe with size 34.31 KB
Process terminated with status 0 (0 minute(s), 0 second(s))
0 error(s), 0 warning(s) (0 minute(s), 0 second(s))
```

C:\Proyectos-CodeBlocks\fechas\bin\Debug\fechas.exe

```
Fecha (dd/mm/aaaa): 1111111/2222222222
1111111/-2072745074/2147303424
Fecha (dd/mm/aaaa): -1/-1/-4
-1/-1/-004
Fecha (dd/mm/aaaa): 11/2/2018
11/02/2018
Fecha (dd/mm/aaaa):
```



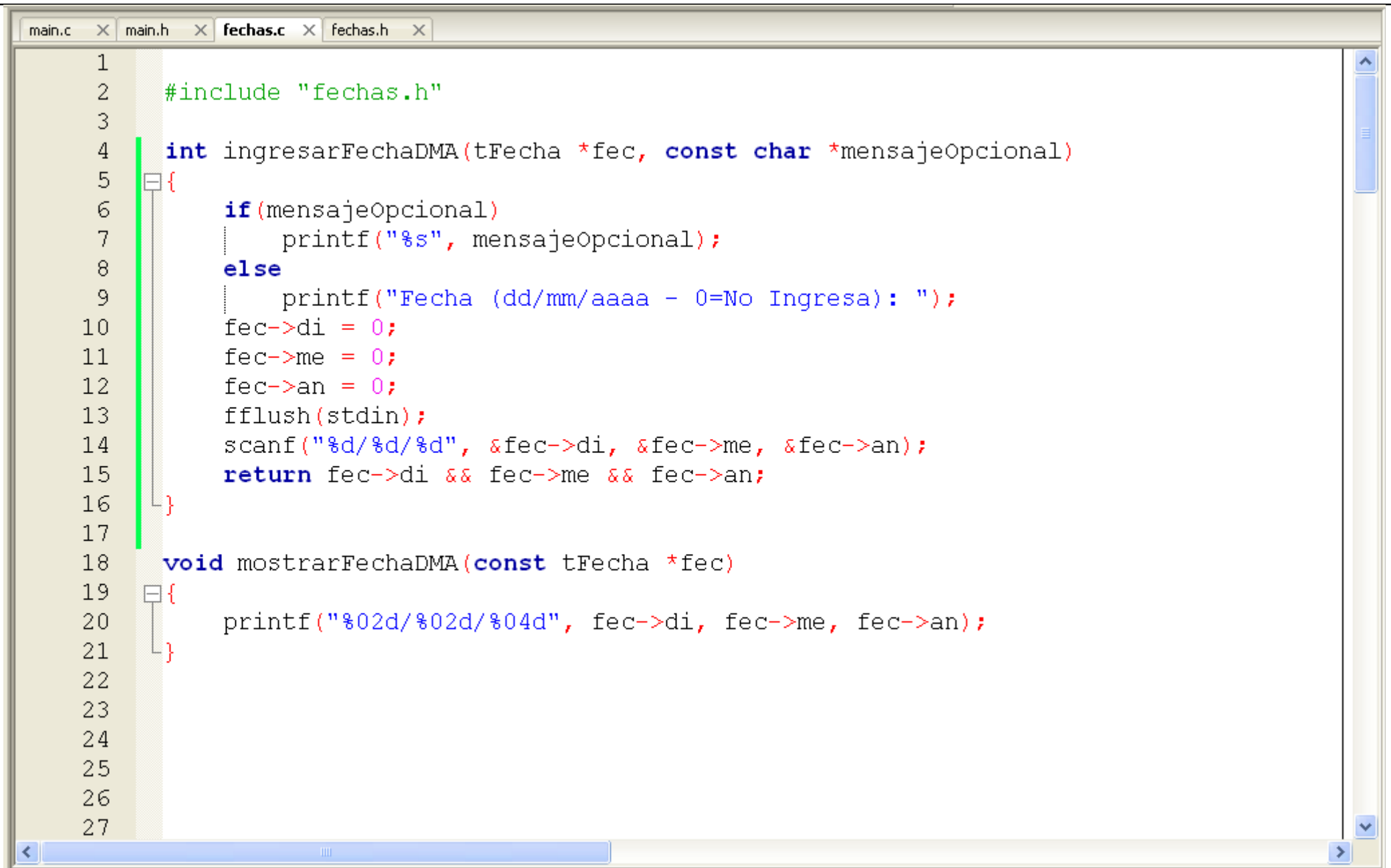
The image shows a code editor window with four tabs: `main.c`, `main.h`, `fechas.c`, and `fechas.h`. The `fechas.c` tab is active, displaying a C program. The code defines two functions: `ingresarFechaDMA` and `mostrarFechaDMA`. The `ingresarFechaDMA` function prompts the user to enter a date (dd/mm/yyyy) and reads the input using `scanf`. The `mostrarFechaDMA` function displays the date using `printf` with zero-padding for the day and month. The code is as follows:

```
1
2  #include "fechas.h"
3
4  int ingresarFechaDMA(tFecha *fec)
5  {
6      printf("Fecha (dd/mm/aaaa) - 0=No Ingresa): ");
7      fec->di = 0;
8      fec->me = 0;
9      fec->an = 0;
10     fflush(stdin);
11     scanf("%d/%d/%d", &fec->di, &fec->me, &fec->an);
12     return fec->di && fec->me && fec->an;
13 }
14
15 void mostrarFechaDMA(const tFecha *fec)
16 {
17     printf("%02d/%02d/%04d", fec->di, fec->me, fec->an);
18 }
19
20
21
22
23
24
25
26
27
```

C:\Proyectos-CodeBlocks\fechas\bin\Debug\fechas.exe

```
Fecha (dd/mm/aaaa) - 0=No Ingresa): 11111111/22222222/33333333
11111111/22222222/33333333
Fecha (dd/mm/aaaa) - 0=No Ingresa): 11111111/22222222/33333333
11111111/22222222/-961633963
Fecha (dd/mm/aaaa) - 0=No Ingresa): 11/2/2020
11/02/2020
Fecha (dd/mm/aaaa) - 0=No Ingresa): 111
Quiso terminar
```

```
Process returned 0 (0x0)   execution time : 117.891 s
Press any key to continue.
```



The image shows a code editor window with four tabs: `main.c`, `main.h`, `fechas.c` (active), and `fechas.h`. The code in `fechas.c` is as follows:

```
1
2  #include "fechas.h"
3
4  int ingresarFechaDMA(tFecha *fec, const char *mensajeOpcional)
5  {
6      if(mensajeOpcional)
7          printf("%s", mensajeOpcional);
8      else
9          printf("Fecha (dd/mm/aaaa - 0=No Ingresa): ");
10     fec->di = 0;
11     fec->me = 0;
12     fec->an = 0;
13     fflush(stdin);
14     scanf("%d/%d/%d", &fec->di, &fec->me, &fec->an);
15     return fec->di && fec->me && fec->an;
16 }
17
18 void mostrarFechaDMA(const tFecha *fec)
19 {
20     printf("%02d/%02d/%04d", fec->di, fec->me, fec->an);
21 }
22
23
24
25
26
27
```



```
main.c x main.h x fechas.c x fechas.h x
9      *      otro orden.
10     **/
11     typedef struct
12     {
13         int di,
14             me,
15             an;
16     } tFecha;
17
18     /**
19     *   función booleana que permite el ingreso de tres enteros para el día, mes
20     *   y año sin garantía de que correspondan a una fecha.
21     *   contempla el caso en que se quiera mostrar un mensaje distinto del
22     *   mensaje por defecto el que se mostrará si recibe NULL.
23     *   contempla el caso en que no se quiera mostrar ningún mensaje cuando reciba
24     *   una cadena vacía (o con blancos o tabulaciones), en este caso, de ser
25     *   necesario, se mostrará el mensaje antes de invocarla.
26     *   contempla el caso en que no se quiera ingresar nada, por ejemplo con
27     *   cero en alguno de los enteros, con lo que devolverá 0 (cero)
28     */
29     int ingresarFechaDMA(tFecha *fec, const char *mensajeOpcional);
30
31     /**
32     *   muestra una fecha en el formato dia/mes/año
33     */
34     void mostrarFechaDMA(const tFecha *fec);
35
```

*main.c x main.h x fechas.c x fechas.h x

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27

```
#include "main.h"

int main()
{
    tFecha fec;

    while(ing)
    {
        ingresarFechaDMA(): int
    }

    puts("Quiso terminar");

    return 0;
}
```

int ingresarFechaDMA(tFecha* fec, const char* mensajeOpcional)
(function)
Description:
función booleana que permite el ingreso de tres enteros para el día, mes y año sin garantía de que correspondan a una fecha. contempla el caso en que se quiera mostrar un mensaje distinto del mensaje por defecto el que se mostrará si recibe NULL. contempla el caso en que no se quiera mostrar ningún mensaje cuando reciba una cadena vacía (o con blancos o tabulaciones), en este caso, de ser necesario, se mostrará el mensaje antes de invocarla. contempla el caso en que no se quiera ingresar nada, por ejemplo con cero en alguno de los enteros, con lo que devolverá 0 (cero) función booleana que permite el ingreso de tres enteros para el día, mes y año sin garantía de que correspondan a una fecha. contempla el caso en que se quiera mostrar un mensaje distinto del mensaje por defecto el que se mostrará si recibe NULL. contempla el caso en que no se quiera mostrar ningún mensaje cuando reciba una cadena vacía (o con blancos o tabulaciones), en este caso, de ser necesario, se mostrará el mensaje antes de invocarla. contempla el caso en que no se quiera ingresar nada, por ejemplo con cero en alguno de los enteros, con lo que devolverá 0 (cero)

The screenshot shows the Code::Blocks IDE interface. The top pane displays the source code for `main.c`, which includes `main.h` and contains a `main` function. The function calls `ingresarFechaDMA` in a loop, `mostrarFechaDMA`, and prints a termination message before returning 0. The bottom pane, titled "Logs & others", shows the build log with the following content:

```
mingw32-gcc.exe -Wall -g -c C:\Proyectos-CodeBlocks\fechas\fechas.c -o obj\Debug\fechas.o
mingw32-gcc.exe -Wall -g -c C:\Proyectos-CodeBlocks\fechas\main.c -o obj\Debug\main.o
mingw32-g++.exe -o bin\Debug\fechas.exe obj\Debug\fechas.o obj\Debug\main.o
Output file is bin\Debug\fechas.exe with size 34.31 KB
Process terminated with status 0 (0 minute(s), 0 second(s))
0 error(s), 0 warning(s) (0 minute(s), 0 second(s))
```

C:\Proyectos-CodeBlocks\fechas\bin\Debug\fechas.exe

Fecha (dd/mm/aaaa - 0=No Ingresa): 22/11/2018

22/11/2018

Fecha (dd/mm/aaaa - 0=No Ingresa): 22/11/-018

22/11/-018

Fecha (dd/mm/aaaa - 0=No Ingresa): 22/11/-2018

22/11/-2018

Fecha (dd/mm/aaaa - 0=No Ingresa): 22/ 11/ 2018

22/11/2018

Fecha (dd/mm/aaaa - 0=No Ingresa): 22 1 2018

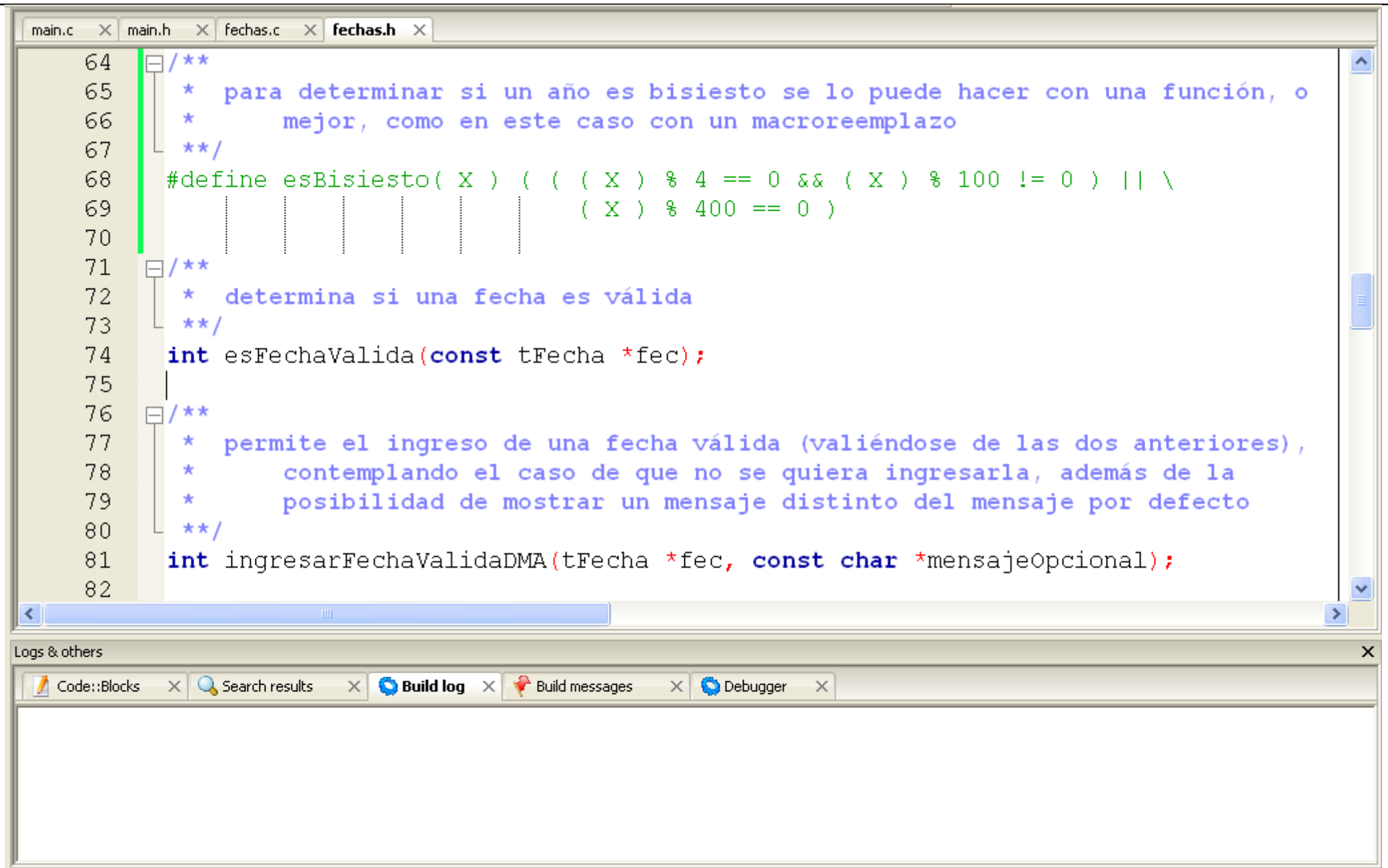
Quiso terminar

Process returned 0 (0x0) execution time : 107.063 s

Press any key to continue.

—

```
main.c x main.h x fechas.c x fechas.h x
36  /**
37  *  Vigencia del Calendario Gregoriano
38  *  Al jueves -juliano- 4 de octubre de 1582 le sucede el
39  *  viernes -gregoriano- 15 de octubre de 1582.
40  *  Diez días desaparecen debido a que ya se habían contado de más en el
41  *  calendario juliano.
42  *  Si hubiera habido calendario gregoriano:
43  *  01/01/1582 -> VIERNES
44  *  01/01/1581 -> JUEVES
45  *  01/01/1580 -> MARTES
46  *  ...
47  *  tomaremos como mínimo válido el año 1 (uno), teniendo en cuenta que no
48  *  existió el año 0 (cero), 'extrapolando' la vigencia del calendario
49  *  gregoriano (su vigencia es -en España, Italia y Portugal, a saber, a
50  *  partir del 15 de octubre de 1582)
51  *  por no ser adoptado en forma universal, su vigencia depende del país
52  *  NOTA: se ponen topes mínimo y máximo, y nuestros algoritmos quedarán
53  *  abiertos a modificaciones (que hasta ahora están más allá)
54  *  OTRA: a partir de su vigencia, por una pequeña diferencia, cada 3623 años
55  *  habrá un día más y para corregirlo, se dejará de contar un bisiesto
56  *  cada 3000 años
57  *  esto quedará como tema abierto a quién tenga intereses muy trascendentes, y
58  *  deberá tener en cuenta determinaciones astronómicas más precisas
59  *  si extrapolamos dentro de un rango razonable nos queda...
60  **/
61  #define      AN_MIN      1
62  #define      AN_MAX      5000
```



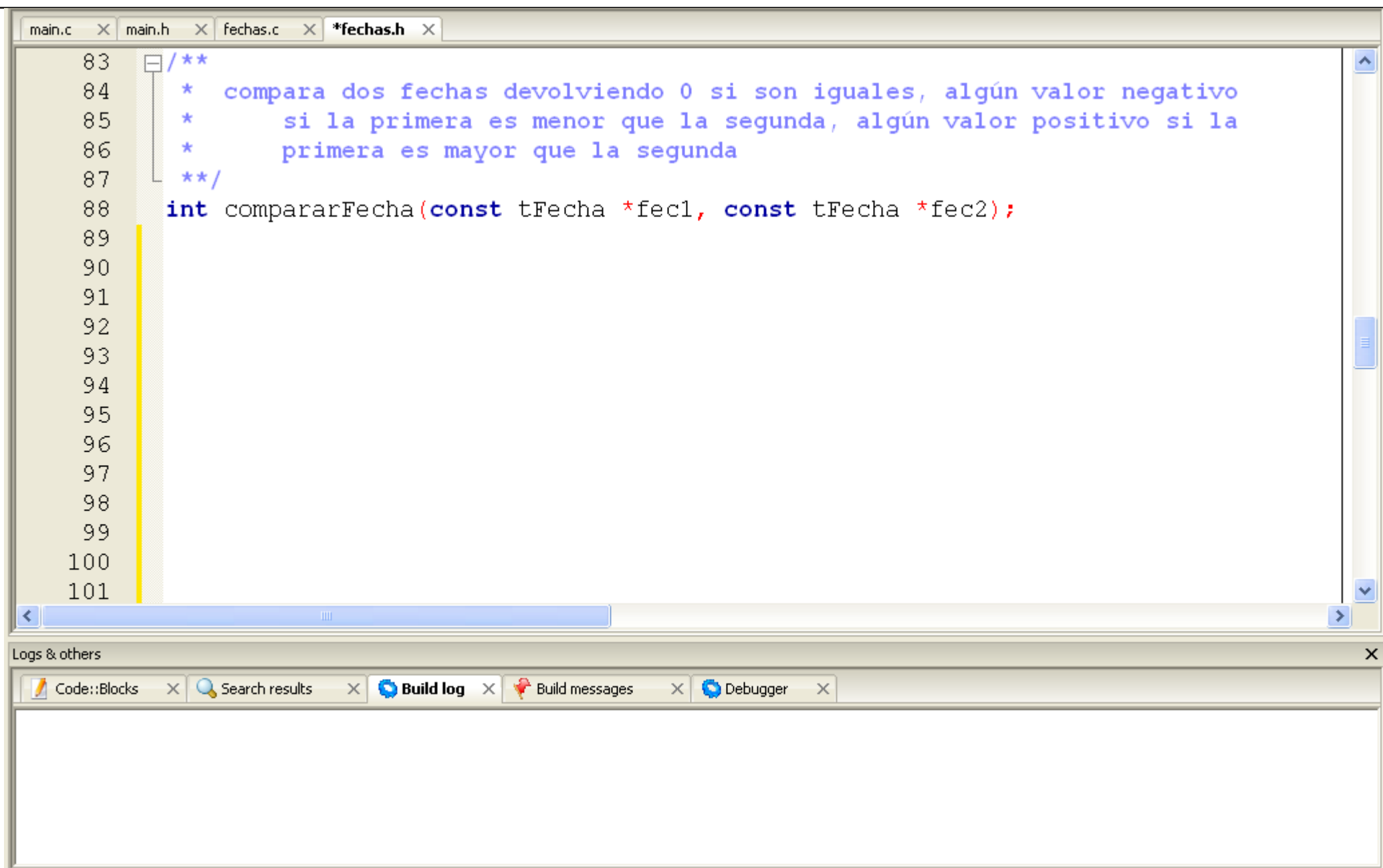
The image shows a screenshot of the Code::Blocks IDE. The top window displays the 'fechas.h' header file. The code includes comments in Spanish explaining the purpose of the functions and a macro for checking leap years. Two functions are declared: 'esFechaValida' and 'ingresarFechaValidaDMA'. The bottom panel shows the 'Logs & others' window with tabs for 'Code::Blocks', 'Search results', 'Build log', 'Build messages', and 'Debugger'.

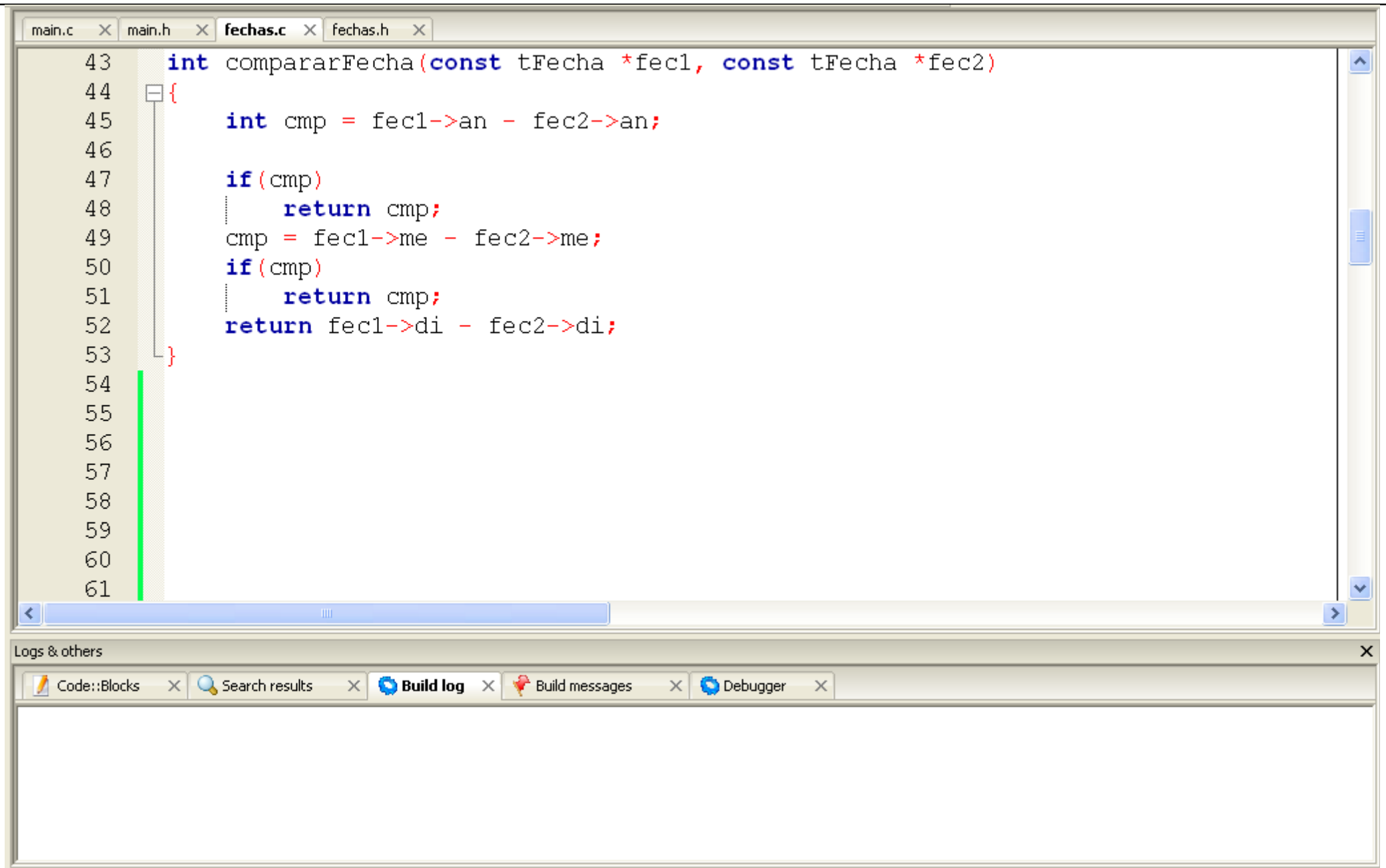
```
64  /**
65  *   para determinar si un año es bisiesto se lo puede hacer con una función, o
66  *       mejor, como en este caso con un macroreemplazo
67  */
68  #define esBisiesto( X ) ( ( ( X ) % 4 == 0 && ( X ) % 100 != 0 ) || \
69  |                         |                         |                         |
70  |                         |                         |                         |
71  |                         |                         |                         |
72  |                         |                         |                         |
73  |                         |                         |                         |
74  |                         |                         |                         |
75  |                         |                         |                         |
76  /**
77  *   permite el ingreso de una fecha válida (valiéndose de las dos anteriores),
78  *       contemplando el caso de que no se quiera ingresarla, además de la
79  *       posibilidad de mostrar un mensaje distinto del mensaje por defecto
80  */
81  int ingresarFechaValidaDMA(tFecha *fec, const char *mensajeOpcional);
82
```

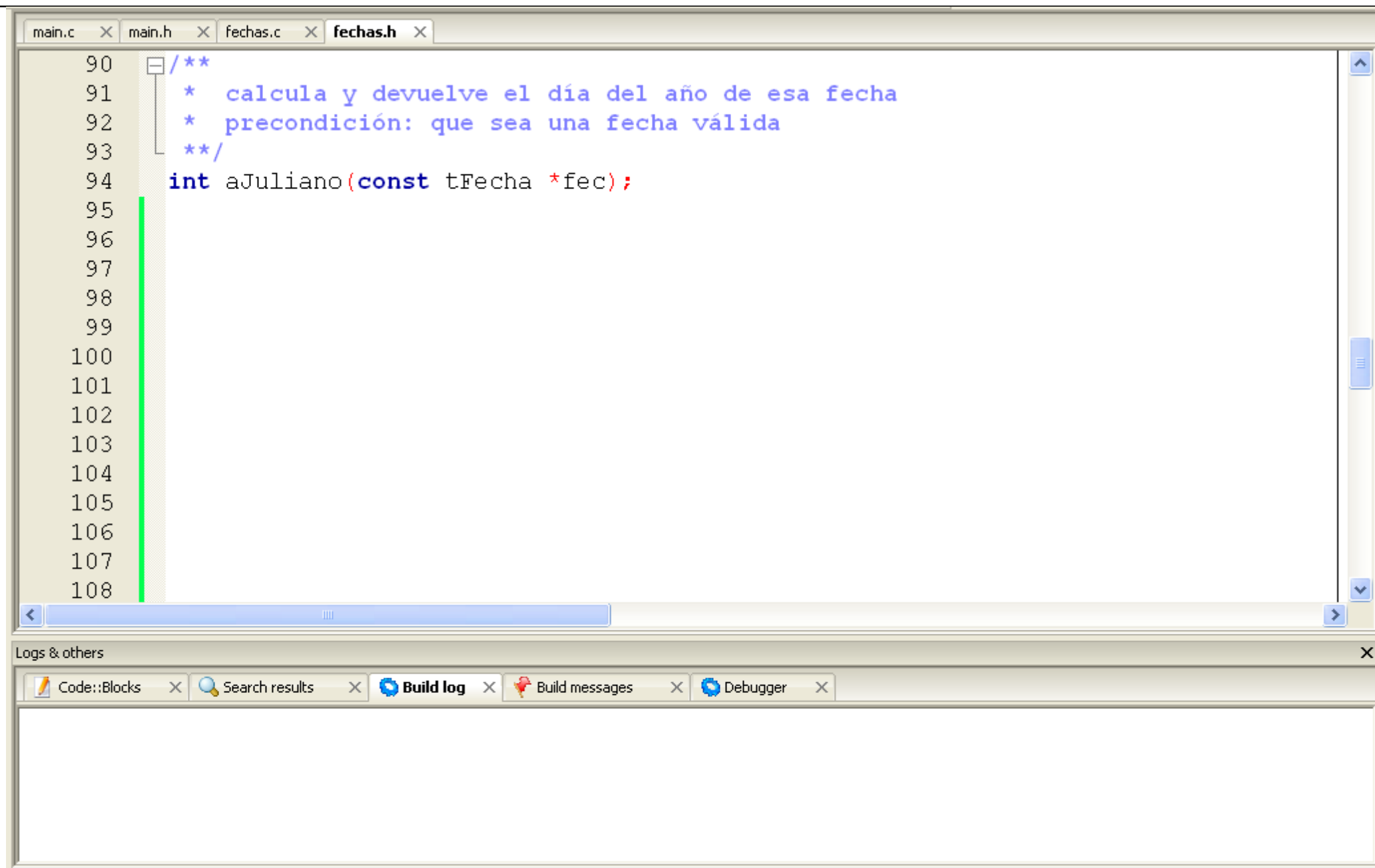
```
main.c x main.h x fechas.c x fechas.h x
23 int esFechaValida(const tFecha *fec)
24 {
25     static const char dias[][12] = {
26         { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 },
27         { 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 } };
28     return fec->me > 0 && fec->me <= 12 &&
29         fec->an >= AN_MIN && fec->an <= AN_MAX &&
30         fec->di > 0 && fec->di <= dias[esBisiesto(fec->an)][fec->me - 1];
31 }
32
33 int ingresarFechaValidaDMA(tFecha *fec, const char *mensajeOpcional)
34 {
35     do
36     {
37         if(!ingresarFechaDMA(fec, mensajeOpcional))
38             return 0;
39     } while(!esFechaValida(fec));
40     return 1;
41 }
```

Logs & others

Code::Blocks x Search results x Build log x Build messages x Debugger x





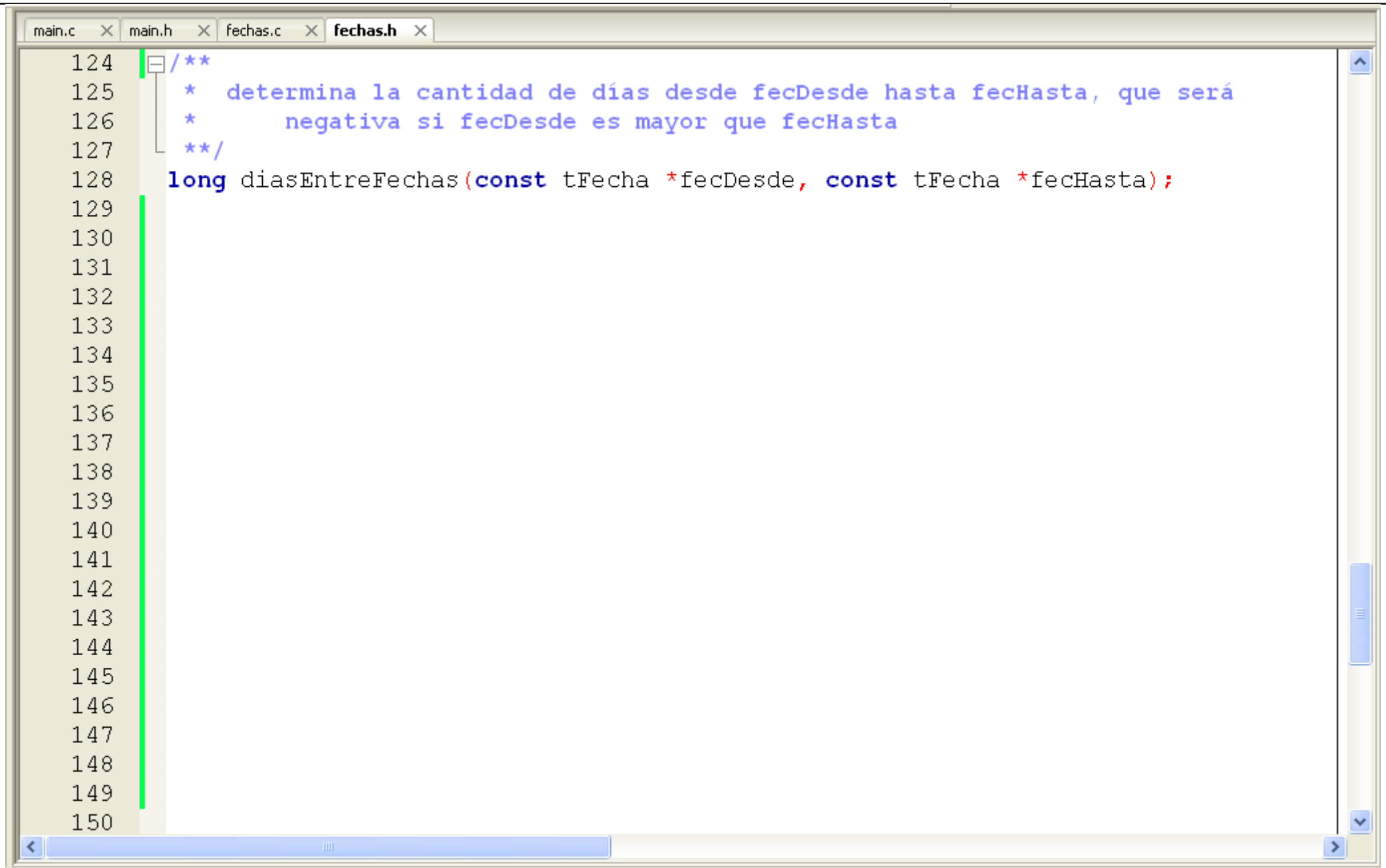


```
main.c x main.h x fechas.c x fechas.h x
96  /**
97  *   la convención, más comunmente utilizada, adoptada a continuación se puede
98  *       alterar a gusto del programador, a costa de apartarse del estándar
99  */
100 #define DOMINGO          0
101 #define LUNES            1
102 #define MARTES           2
103 #define MIERCOLES        3
104 #define JUEVES           4
105 #define VIERNES          5
106 #define SABADO           6
107
108  /**
109  *   determina el número de día de la semana de una fecha
110  *   se tiene en cuenta que:
111  *   -   en un lapso de 400 años la cantidad de días es múltiplo de 7
112  *   -   para cada uno de los años consecutivos de esos intervalos el primero de
113  *       enero de cada año comienza en el mismo día
114  *   -   el primero de enero de los años múltiplo de 400 es  SABADO  (6)
115  *   tomamos como año base el año múltiplo de 400 inmediato anterior al año
116  *       pero sólo es necesario calcular la distancia al año base en años
117  *   por cada año se suma 1 al nro de día del año base, salvo que
118  *       cada cuatro años es bisiesto y se suma uno más, pero cada 100 años
119  *       no es bisiesto y se resta 1.
120  *   si el año no fuera el año base calculado, se suma 1 porque es bisiesto
121  */
122  int nroDeDiaDeLaSemana(const tFecha *fec);
```

The image shows a code editor window with four tabs: main.c, main.h, *fechas.c, and fechas.h. The active tab is *fechas.c. The code is written in C and defines a function nroDeDiaDeLaSemana. The function takes a constant pointer to a tFecha structure and returns an integer representing the day of the week. The code is as follows:

```
83  int nroDeDiaDeLaSemana(const tFecha *fec)
84  {
85      int distBase = fec->an % 400;
86
87      return ( SABADO - 1 + distBase + distBase / 4 - distBase / 100 +
88              (distBase != 0) + aJuliano(fec) ) % 7;
89  }
```

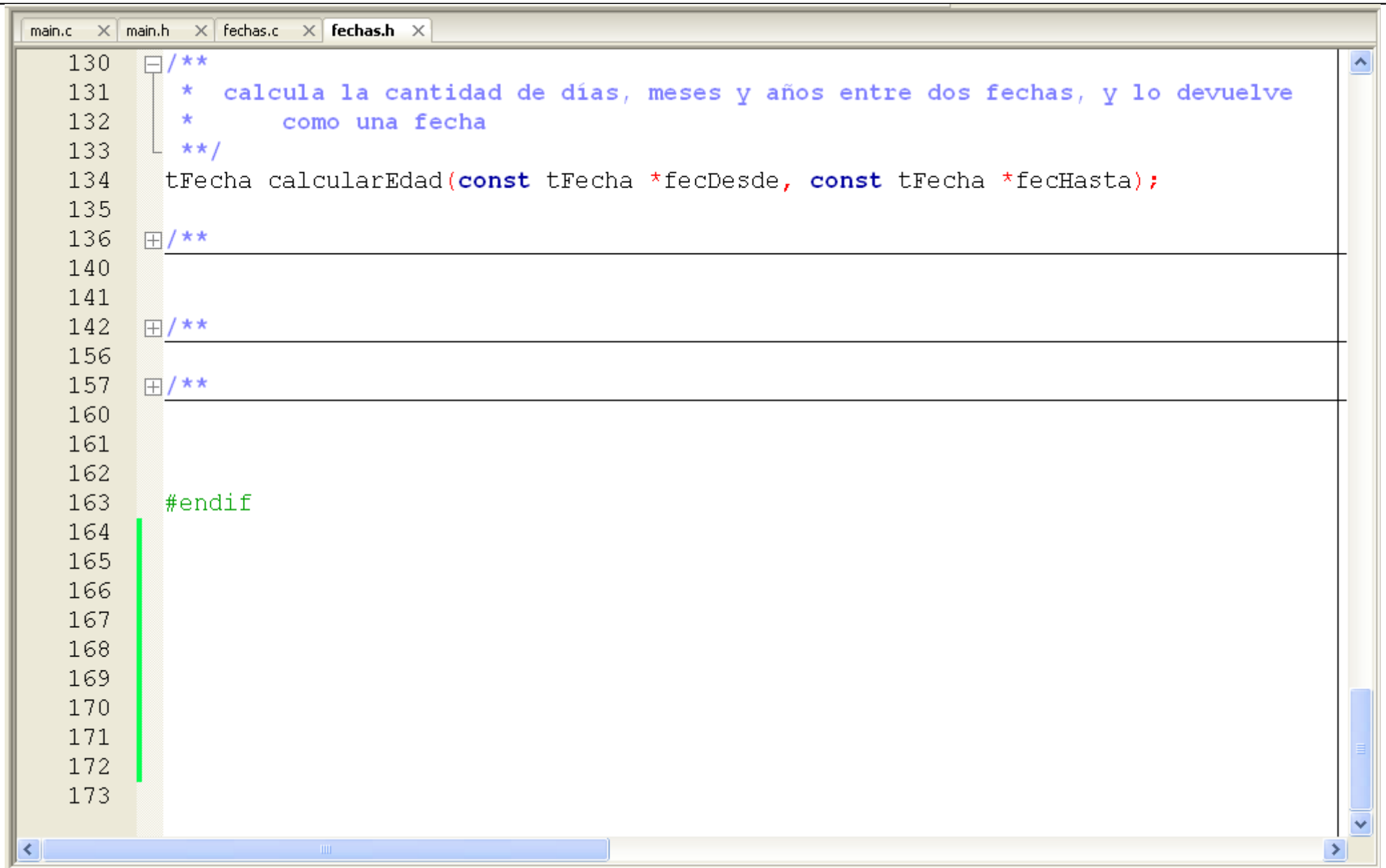
The line numbers 83 through 109 are visible on the left side of the editor. A yellow vertical bar is positioned at line 90. The editor has a scrollbar on the right and a horizontal scrollbar at the bottom.



The image shows a code editor window with four tabs: 'main.c', 'main.h', 'fechas.c', and 'fechas.h'. The 'fechas.h' tab is active. The code is written in C and defines a function 'diasEntreFechas'. The function takes two pointers to a 'tFecha' struct as arguments and returns a 'long' value. The code is as follows:

```
124  /**
125   *   determina la cantidad de días desde fecDesde hasta fecHasta, que será
126   *       negativa si fecDesde es mayor que fecHasta
127   */
128  long diasEntreFechas(const tFecha *fecDesde, const tFecha *fecHasta);
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
```

```
main.c x main.h x fechas.c x fechas.h x
72 long diasEntreFechas(const tFecha *fecDesde, const tFecha *fecHasta)
73 {
74     int anBase = fecDesde->an <= fecHasta->an ? fecDesde->an : fecHasta->an,
75         distBaseDesde = fecDesde->an - anBase,
76         distBaseHasta = fecHasta->an - anBase;
77     long diasBaseAHasta = distBaseHasta * 365L + distBaseHasta / 4 -
78         distBaseHasta / 100 + distBaseHasta / 400 +
79         (distBaseHasta != 0) + aJuliano(fecHasta),
80         diasBaseADesde = distBaseDesde * 365L + distBaseDesde / 4 -
81         distBaseDesde / 100 + distBaseDesde / 400 +
82         (distBaseDesde != 0) + aJuliano(fecDesde);
83     return diasBaseAHasta - diasBaseADesde;
84 }
85
86
87
88
89
90
91
92
93
94
95
96
97
98
```



The image shows a code editor window with four tabs: 'main.c', 'main.h', 'fechas.c', and 'fechas.h'. The 'fechas.h' tab is active. The code is as follows:

```
130  /**
131      *   calcula la cantidad de días, meses y años entre dos fechas, y lo devuelve
132      *       como una fecha
133      */
134  tFecha calcularEdad(const tFecha *fecDesde, const tFecha *fecHasta);
135
136  /**
140
141
142  /**
156
157  /**
160
161
162
163  #endif
164
165
166
167
168
169
170
171
172
173
```

The code blocks between lines 136-140, 142-156, and 157-160 are collapsed, indicated by a minus sign in the left margin. A vertical green line is positioned at the start of line 164. The editor includes a scrollbar on the right and a status bar at the bottom.

```
main.c x main.h x fechas.c x fechas.h x
86 tFecha calcularEdad(const tFecha *fecDesde, const tFecha *fecHasta)
87 {
88     static const char dias[][13] = {
89         { 0, 31, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30 },
90         { 0, 31, 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30 } };
91     ///      dic ene feb mar abr may jun jul ago set oct nov
92     tFecha edad;
93
94     edad = *fecHasta;
95     if( (edad.di -= fecDesde->di) < 0)
96     {
97         edad.di += dias[esBisiesto(edad.an)][edad.me];
98         edad.me--;
99     }
100     if( (edad.me -= fecDesde->me) < 0)
101     {
102         edad.me += 12;
103         edad.an--;
104     }
105     edad.an -= fecDesde->an;
106     return edad;
107 }
108
109 /**
140
141
142
```