

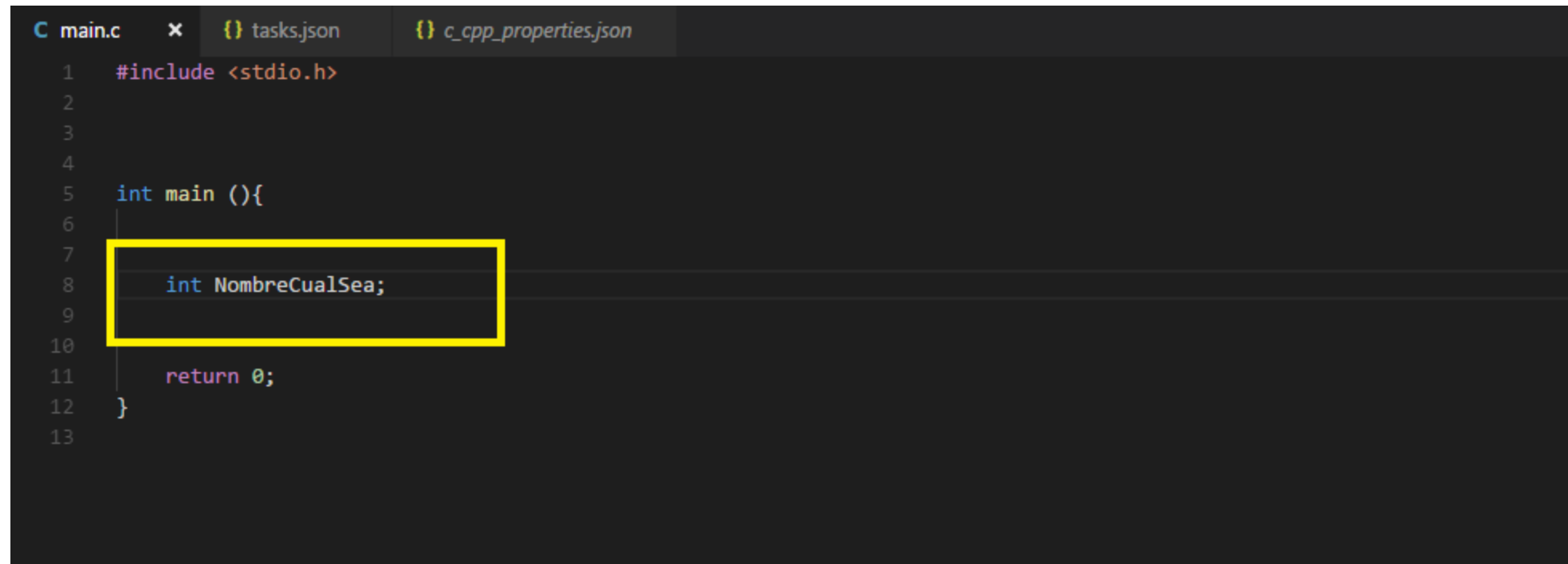
Tipos de Datos

Tipos de Datos

Los tipos de datos son los que definen una variable, estos son necesarios de clasificar para el correcto manejo de operaciones, ellos ocupan un espacio en específico en la memoria medida en bytes. Básicamente existen 5 tipos de datos:

- **Enteros:** Representan los tipos de datos discretos, es decir solo los números que pertenezcan al conjunto de números enteros, estos ocupan en memoria un peso de 4 bytes. Para poder declararlos es necesario poner el nombre de la variable después de la palabra “int”, así pues, el computador podrá reconocer que es un tipo de dato entero.

Tipos de Datos



The image shows a code editor with three tabs: 'main.c', 'tasks.json', and 'c_cpp_properties.json'. The 'main.c' tab is active, displaying a C program. The code is as follows:

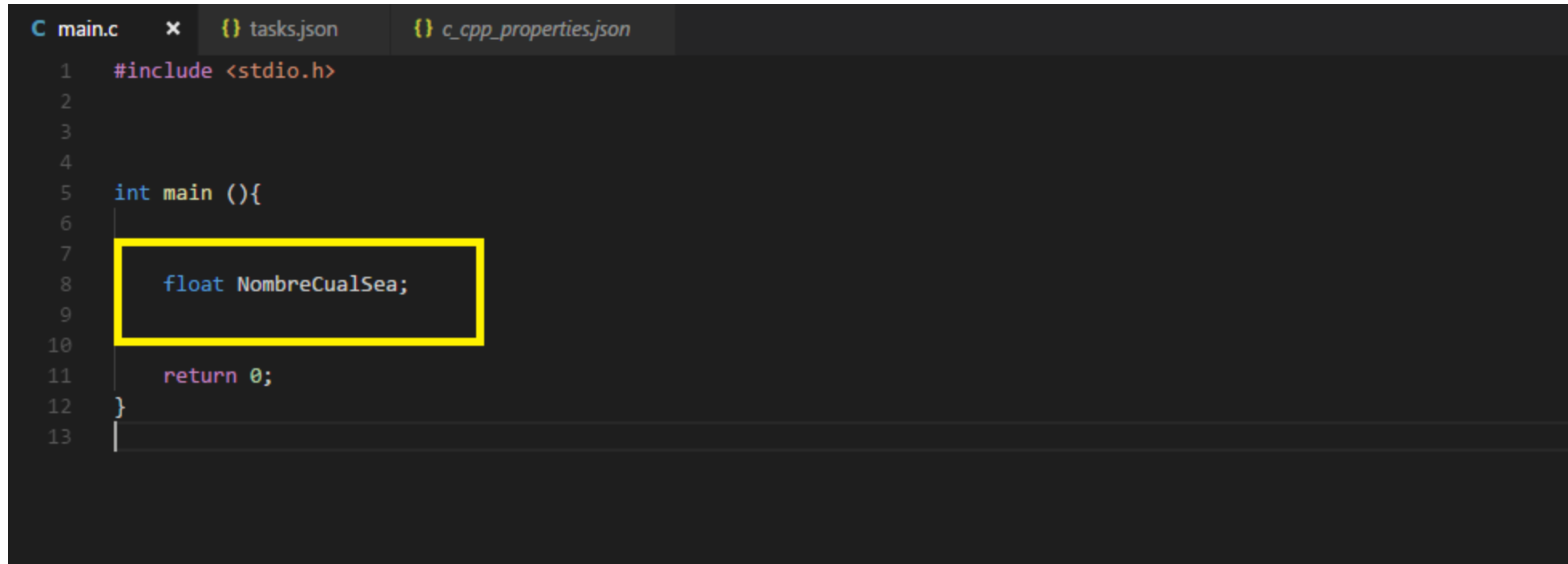
```
1  #include <stdio.h>
2
3
4
5  int main (){
6
7
8      int NombreCualSea;
9
10
11     return 0;
12 }
13
```

The line `int NombreCualSea;` on line 8 is highlighted with a yellow rectangular box.

Tipos de Datos

- **Reales:** Representan los tipos de datos que tienen parte decimal, sin embargo no pueden representar todo los números reales debido a que tiene un máximo de memoria para ocupar. Por lo general se define poniendo “float” antes del nombre, este tendrá un peso de 4 bytes en memoria.

Tipos de Datos



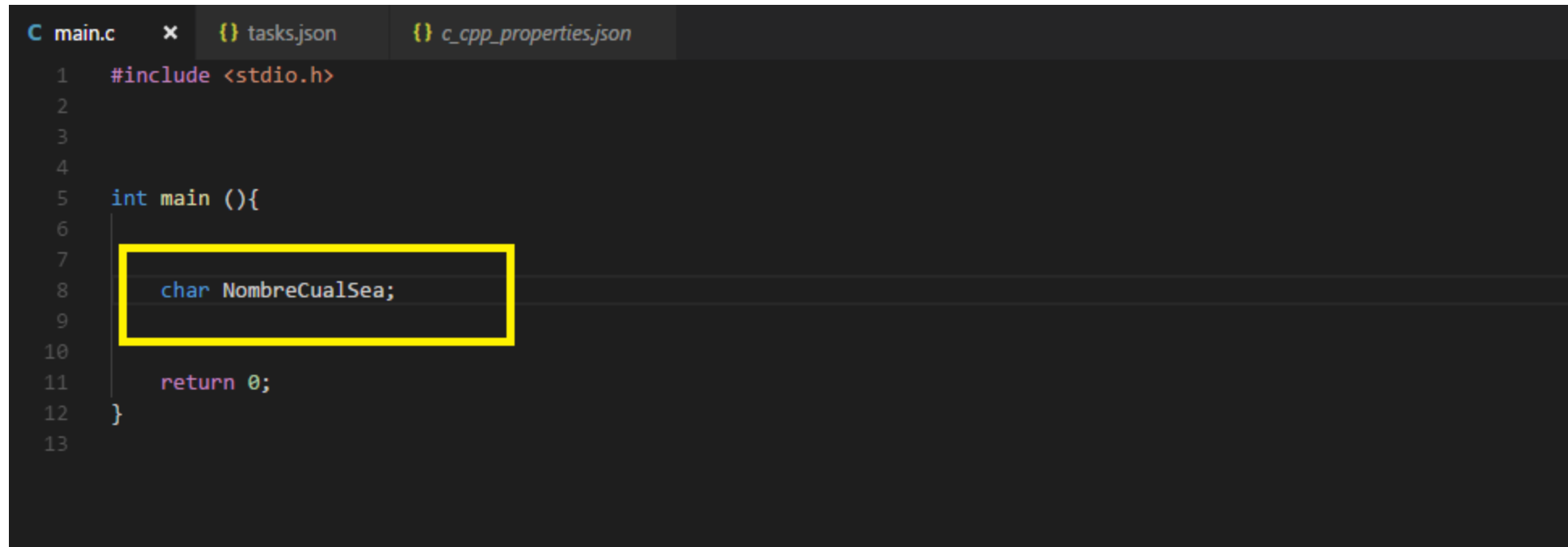
```
C main.c x {} tasks.json {} c_cpp_properties.json
1  #include <stdio.h>
2
3
4
5  int main (){
6
7
8      float NombreCualSea;
9
10
11     return 0;
12 }
13
```

The image shows a code editor window with three tabs: 'main.c', 'tasks.json', and 'c_cpp_properties.json'. The 'main.c' tab is active, displaying a C program. The program includes a header file, a main function, and a variable declaration. The variable declaration 'float NombreCualSea;' is highlighted with a yellow rectangular box. The code is as follows:

Tipos de Datos

- **Caracteres:** Representan los caracteres como letras, paréntesis corchetes, entre otros símbolos pertenecientes al código ASCII el cual cada símbolo tiene como referencia un numero del 0 al 255. Su peso es de 1 byte y para poder declararlo se debe poner “char” antes del nombre.

Tipos de Datos



The image shows a code editor with three tabs: 'main.c', 'tasks.json', and 'c_cpp_properties.json'. The 'main.c' tab is active, displaying a C program. The code is as follows:

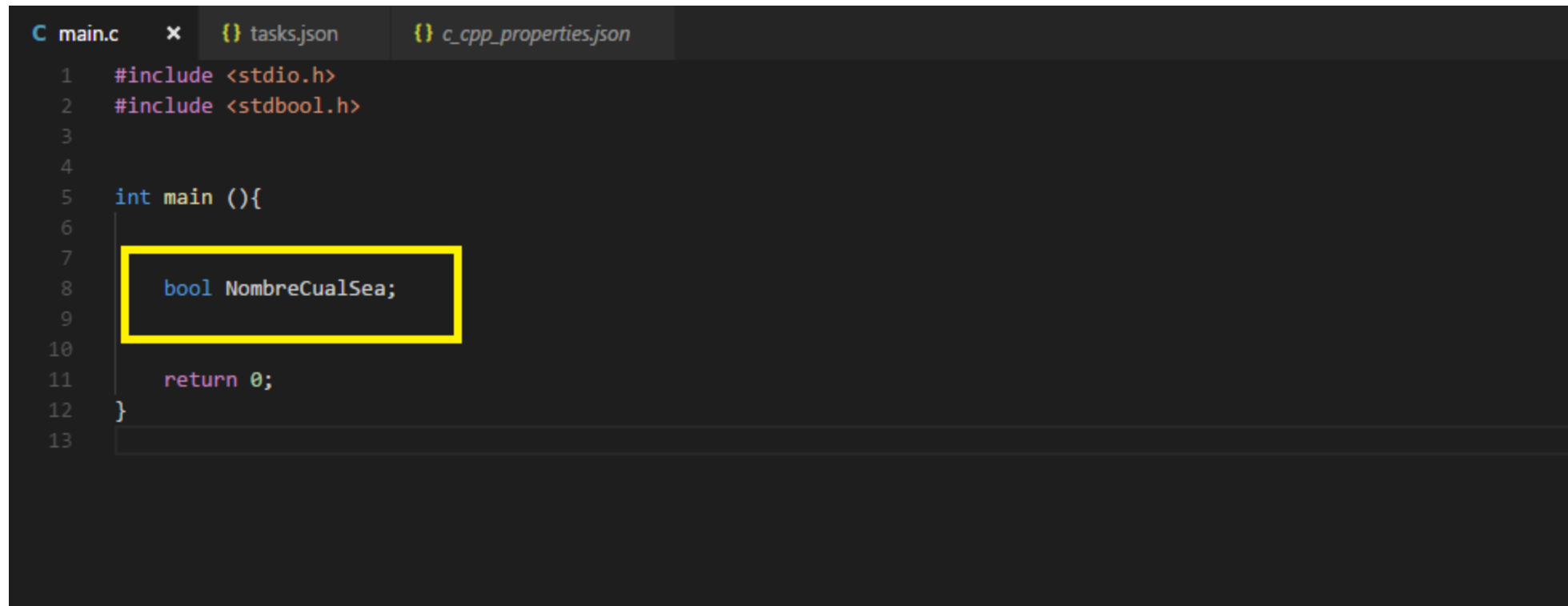
```
1  #include <stdio.h>
2
3
4
5  int main (){
6
7      char NombreCualSea;
8
9
10
11     return 0;
12 }
13
```

The line `char NombreCualSea;` on line 7 is highlighted with a yellow rectangular box, indicating the focus on the `char` data type.

Tipos de Datos

- **Booleano:** Es un tipo de dato que solo tiene dos posibles valores, False (Falso) y True (Verdadero), este también se puede escribir con 0 y 1 respectivamente. El tipo de dato booleano es muy liviano (1 byte) por lo que mayormente se usa como bandera en ciertas secciones de algoritmos que se requiera una respuesta de verdadero o falso.
- Para declararlo solo hace falta poner “bool” seguidamente del nombre de la variable.

Tipos de Datos



```
C main.c x {} tasks.json {} c_cpp_properties.json
1  #include <stdio.h>
2  #include <stdbool.h>
3
4
5  int main (){
6
7      bool NombreCualSea;
8
9
10
11     return 0;
12 }
13
```

The image shows a code editor window with three tabs: 'main.c', 'tasks.json', and 'c_cpp_properties.json'. The 'main.c' tab is active, displaying a C program. The program includes `<stdio.h>` and `<stdbool.h>`. The `main` function is defined, and a boolean variable `NombreCualSea` is declared on line 7. This declaration is highlighted with a yellow rectangular box. The program ends with `return 0;` and a closing brace for the `main` function.

Tipos de Datos

- **Puntero:** es una variable que **contiene la dirección de memoria** de un dato. Esto quiere decir, que el puntero apunta a una dirección virtual donde está el dato o la variable en la memoria física.
- Un puntero puede apuntar a cualquier tipo de dato, tales como entero, flotante, carácter y estructura de datos. Los punteros se pueden utilizar para referencia y manipular estructuras de datos.
- Para poder declararlo se pone primero el tipo de dato que se quiere apuntar y luego un asterisco seguido del nombre.

Tipos de Datos

```
C main.c x {} tasks.json {} c_cpp_properties.json
1  #include <stdio.h>
2  #include <stdbool.h>
3
4
5  int main (){
6
7      int * NombreCualSea;
8
9
10
11     return 0;
12 }
13
```

Tipos de Datos

- **Estructuras:** Las estructuras son agrupaciones de tipos de datos que definen un nuevo tipo de dato. Hay varias formas de definirla pero la mas común seria del siguiente modo:

```
Typdef struct {    } NombreDelaEstructura;
```

- Dentro de ellas se deberá declarar todo tipo de dato que se desee, conforme dicha estructura, estos pueden ser enteros, reales, booleanos, caracteres, arreglos, cadenas e inclusive otras estructuras.
- Una vez declarada para poder usarla deberemos crear una variable la cual sea este tipo de dato. Para poder hacer esto solo hace falta poner el nombre de la estructura y luego el nombre de la variable seguida de un “;”

Tipos de Datos



```
C main.c x {} tasks.json {} c_cpp_properties.json
1  #include <stdio.h>
2  #include <stdbool.h>
3
4
5  typedef struct {
6      int a;
7      float b;
8      char c;
9  } Mi_estructura;
10
11
12
13
14
15  int main (){
16      Mi_estructura NombreCualsea;
17
18      return 0;
19
20
21 }
```