**Guía Formativa de Ejercicios**

**“Librerías E/S”**

Aplicando la metodología de la programación para el análisis de problemas, desarrolle con el lenguaje de programación “C” los siguientes ejercicios. Cada ejercicio deberá ser considerado como un subproceso y/o función. Cada ejercicio puede implementar sus propios subprocesos y/o funciones.

1. **Entrada y Salida (E/S) Estándar en C/C++**

Todo archivo fuente que utilice funciones de E/S deberá contener la línea:

**#include** <**archivo de cabecera**>

El archivo de cabecera puede ser “**stdio.h**”, en el caso de C, o “**iostream.h**”, en el caso de C++. Ambos archivos definen ciertas macros, variables y funciones empleadas para E/S. C++, por ser la extensión de C, puede utilizar ambos archivos.

1. **Funciones de librerías para E/S sin Formato**
2. Complete la tabla, según el ejemplo (1 fila).

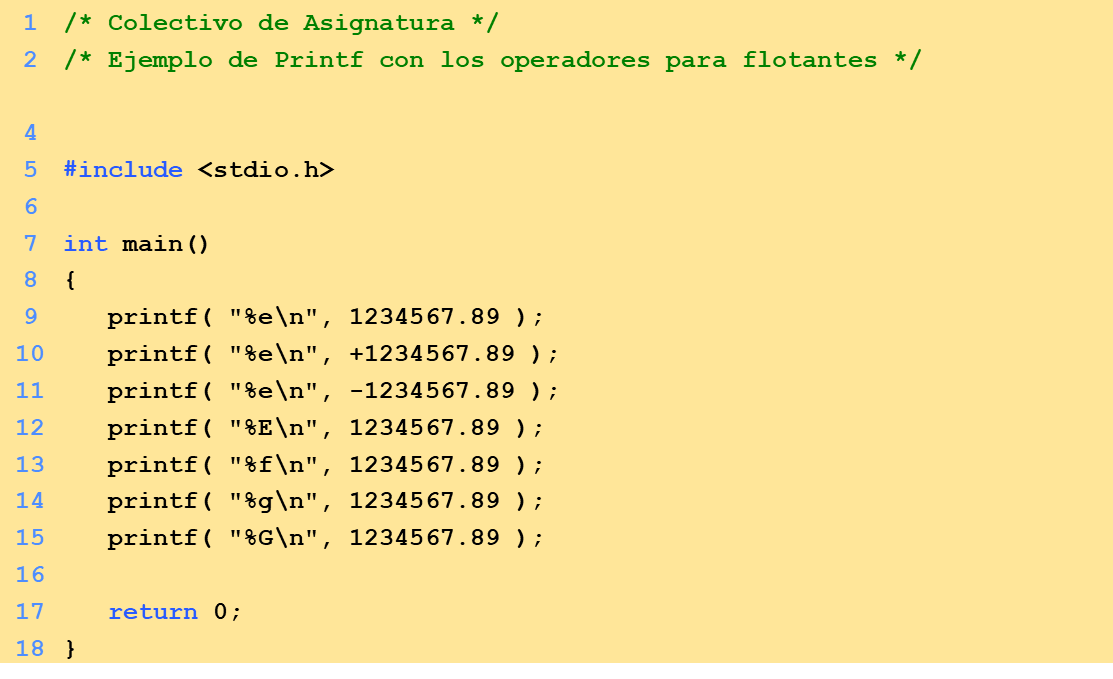
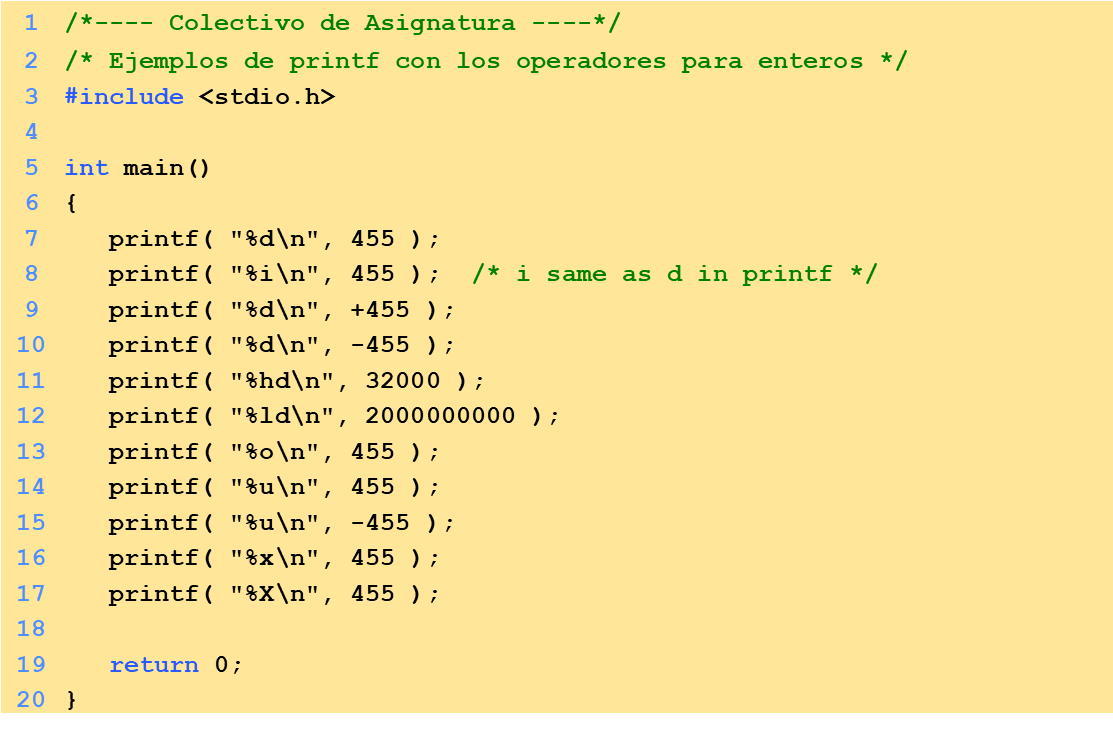
|  |  |  |  |
| --- | --- | --- | --- |
| **Funciones** | **Sintaxis** | **Biblioteca** | **Utilidad** |
| getchar() | Variable\_de\_caracter = getchar();  En donde variable\_de\_caracter es alguna variable de carácter previamente declarada. | stdio.h | Lee un carácter del teclado, espera hasta que se pulsa la tecla <enter> |
| putchar() | La sintaxis es gets(cadena) | stdio.h | se utiliza para leer una cadena de caracteres desde el teclado, hasta que se pulsa la tecla <enter>. Esta función se considera insegura debido a que no hay forma de especificar el tamaño máximo de la cadena que se puede leer |
| gets() |  |  |  |
| puts() |  |  |  |
| fflush(stdin) |  |  |  |
| fflush(stdout) |  |  |  |
| scanf() |  |  |  |
| printf() |  |  |  |
| getline() |  |  |  |

1. **Caracteres de conversión y secuencias de escape para E/S con Formato**
2. Caracteres de conversión y secuencias de escape.

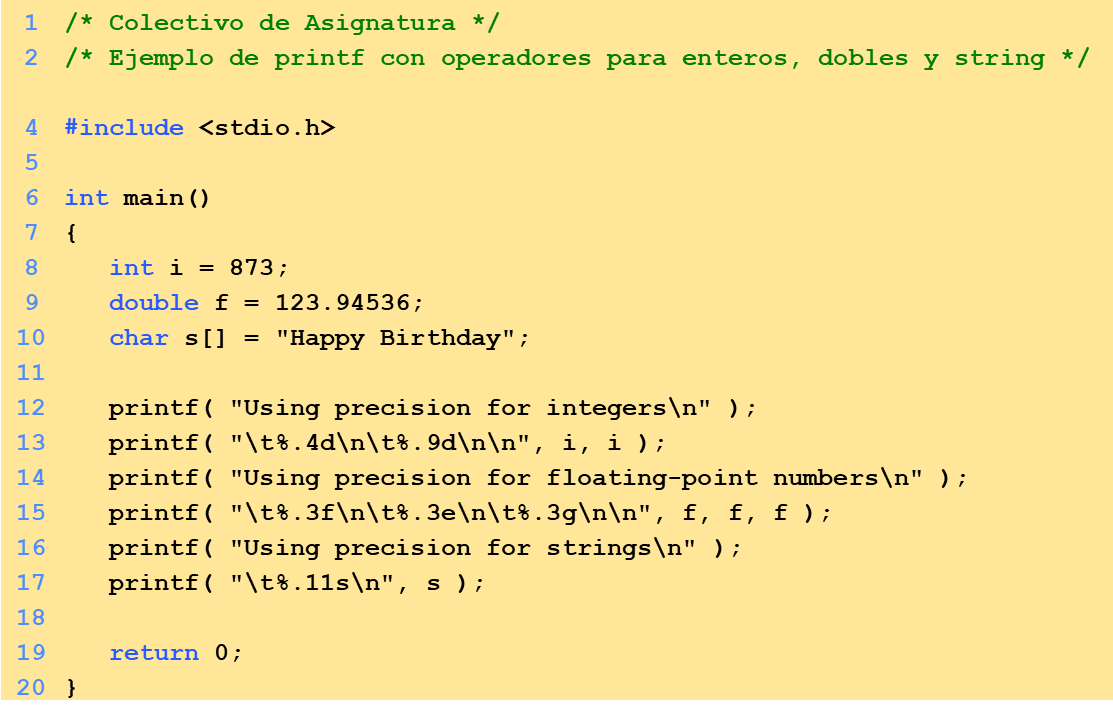
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Carácter de conversión** | **Significado** |  | **Secuencia de**  **Escape** | **Significado** |
| **c** | El dato es un carácter. |  | \b | Espacio atrás |
| **d** | El dato es un entero decimal. |  | \f | Salto de página |
| **e** | El dato es un valor en coma flotante. |  | \n | Salto de línea |
| **f** | El dato es un valor en coma flotante. |  | \r | Retorno de carro |
| **g** | El dato es un valor en coma flotante. |  | \t | Tabulación horizontal |
| **h** | El dato es un entero corto. |  | \v | Tabulación vertical |
| **i** | El dato es un entero decimal, octal o hexadecimal. |  | \\ | Barra invertida |
| **o** | El dato es un entero octal. |  | \’ | Comilla simple |
| **s** | El dato es una cadena de caracteres seguida de un carácter de espaciado. (se añade automáticamente el carácter nulo \0 al final). |  | \” | Comillas dobles |
| **u** | El dato es un entero decimal sin signo. |  | \ooo | Visualiza un carácter cuyo código ASCII es OOO en octal |
| **x** | El dato es un entero hexadecimal. |  | \xhhh | Visualiza un carácter cuyo código ASCII es HHH en hexadecimal |
| **[ . . . ]** | El dato es una cadena de caracteres que puede incluir caracteres espaciados. |  |

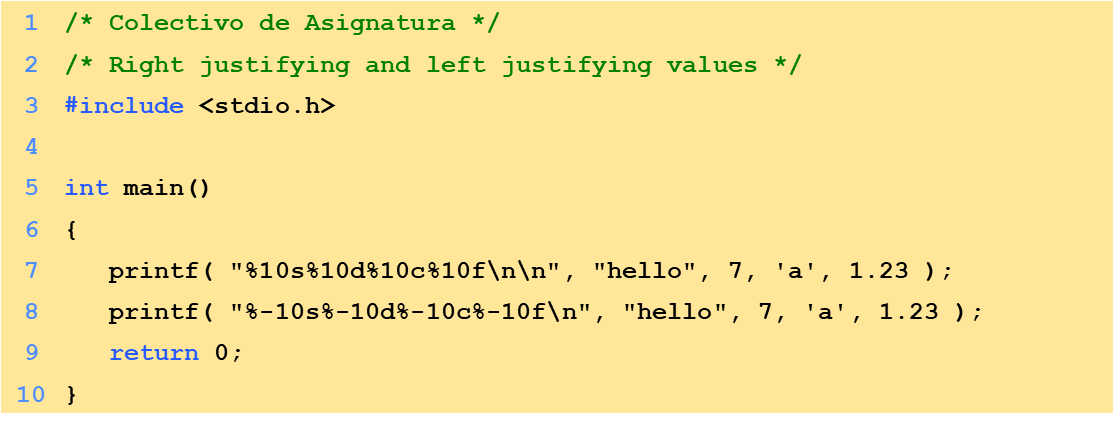
1. **Implementando funciones de librerías para E/S con formato**

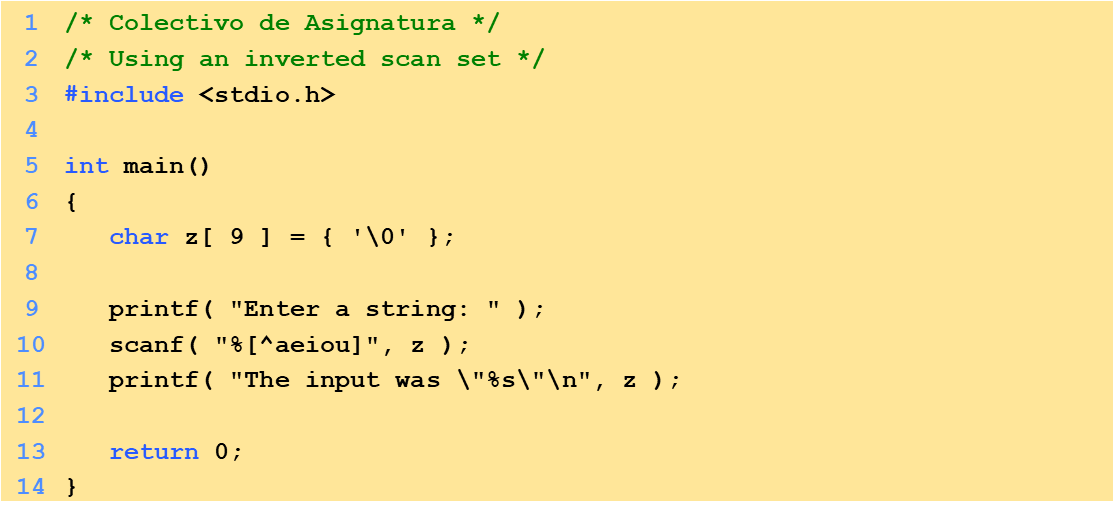
Según la imagen presentada copie el código y compile.

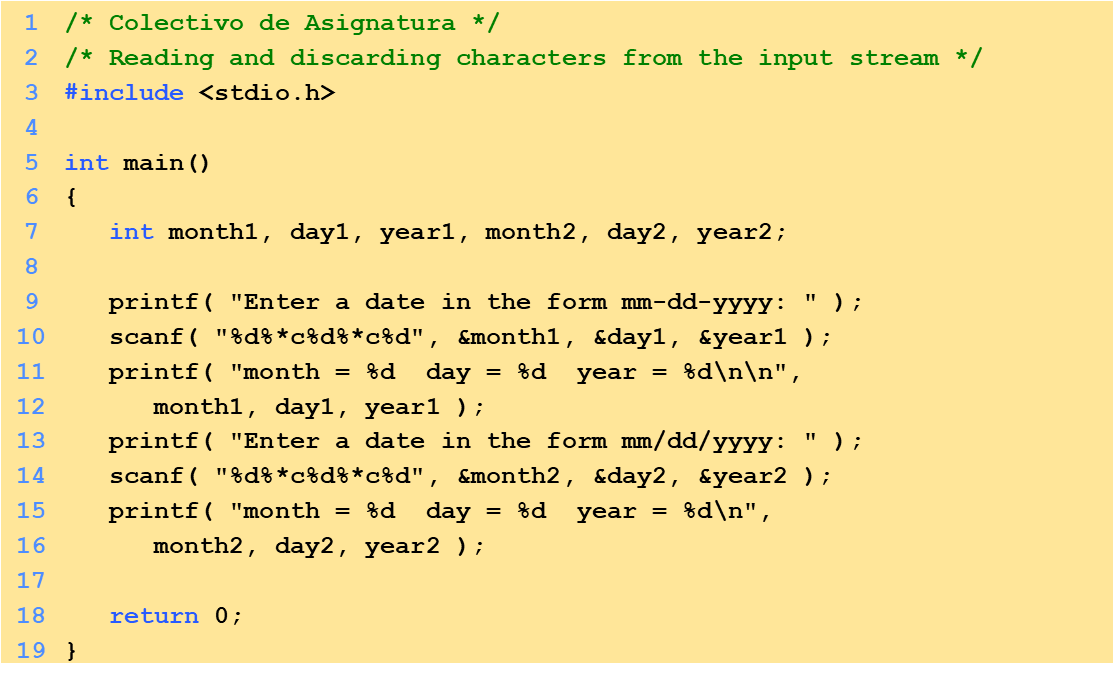
**Ejercicio#1** **Ejercicio #2**

**Ejercicio #3**



 **Ejercicio #4**

**Ejercicio #5**

 **Ejercicio #6**