**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).

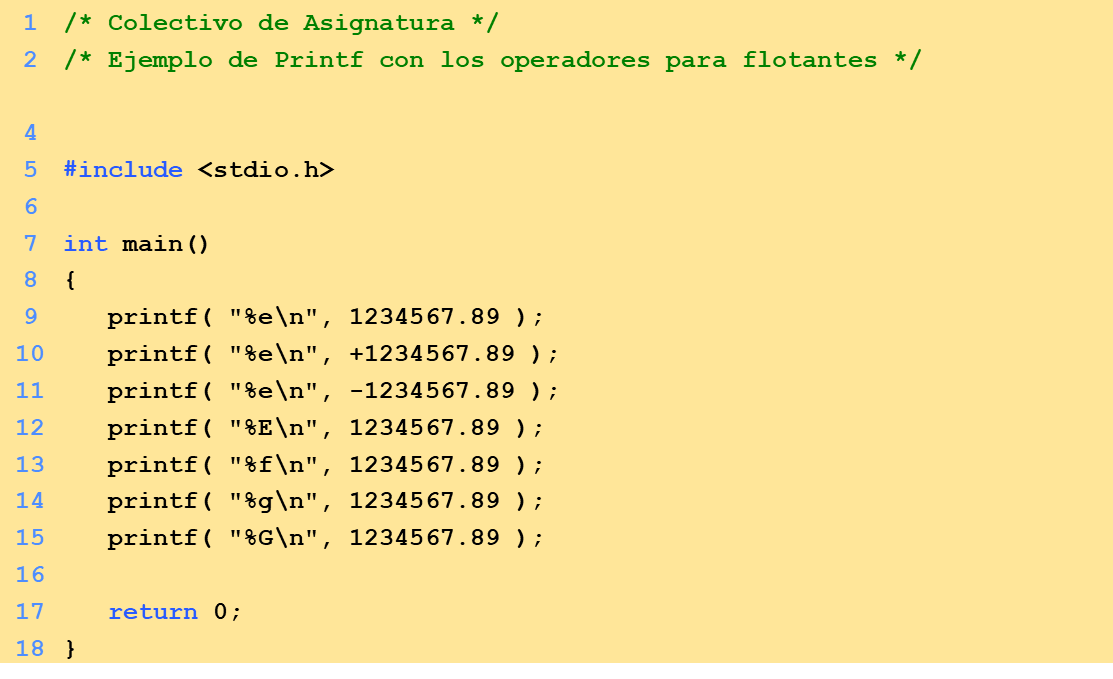
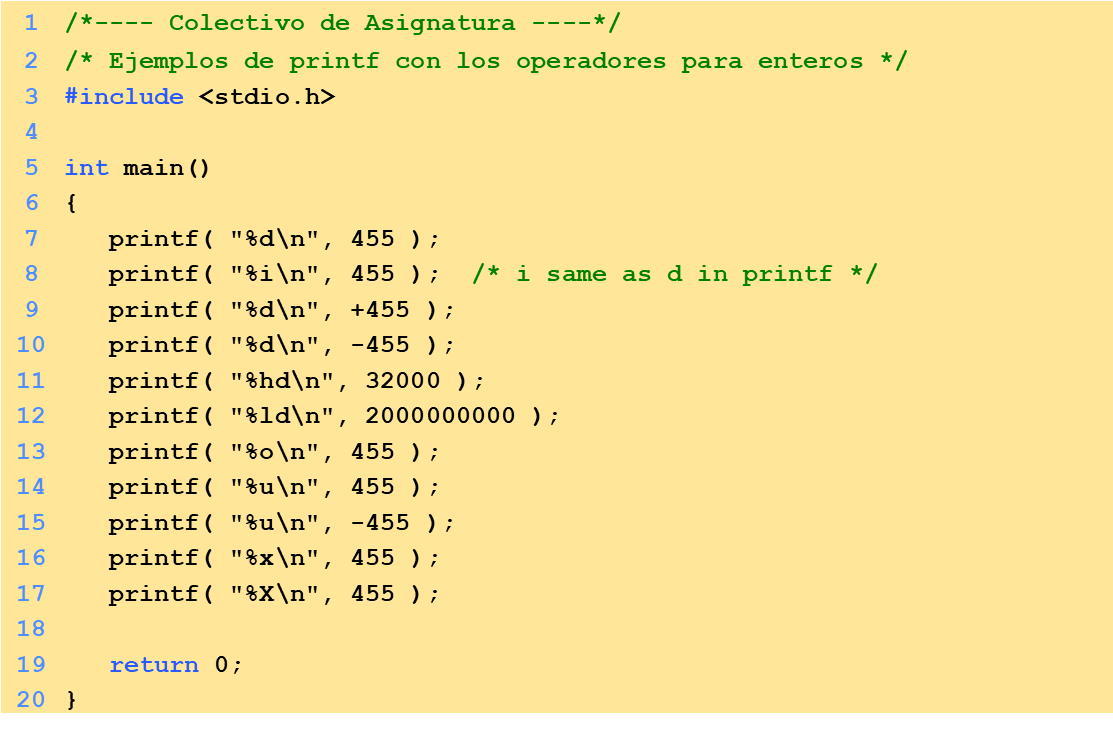
|  |  |  |  |
| --- | --- | --- | --- |
| **Funciones** | **Sintaxis** | **Biblioteca** | **Utilidad** |
| getchar() | Variable\_de\_caracter = getchar();  En donde variable\_de\_caracter es alguna variable de carácter previamente declarada. | stdio.h | Lee un carácter del teclado, espera hasta que se pulsa la tecla <enter> |
| putchar() | putchar(caracter); | stdio.h | Imprime un carácter en la pantalla. |
| gets() | gets(cadena); | stdio.h | Lee una cadena de caracteres desde el teclado y la almacena en la variable cadena. Esta función es peligrosa ya que puede causar un desbordamiento de búfer si la entrada es más grande que el tamaño de la variable cadena. |
| puts() | puts(cadena); | stdio.h | Imprime una cadena de caracteres en la pantalla seguida de un salto de línea. |
| fflush(stdin) | fflush(stdin); | stdio.h | Limpia el búfer de entrada de la consola. Esta función no es estándar y su comportamiento puede variar entre diferentes sistemas. |
| fflush(stdout) | fflush(stdout); | stdio.h | Limpia el búfer de salida de la consola. Esta función asegura que cualquier texto que se haya escrito en la pantalla se muestre inmediatamente. |
| scanf() | scanf(formato, argumentos); | stdio.h | Lee datos desde el teclado en función del formato especificado en la cadena de formato y los almacena en las variables especificadas |
| printf() | printf(formato, argumentos); | stdio.h | Imprime texto formateado en la pantalla. |
| getline() | getline(&cadena, &longitud, archivo); | stdio.h | Lee una línea de un archivo y la almacena en la variable cadena. La longitud de la línea se almacena en la variable longitud. |

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

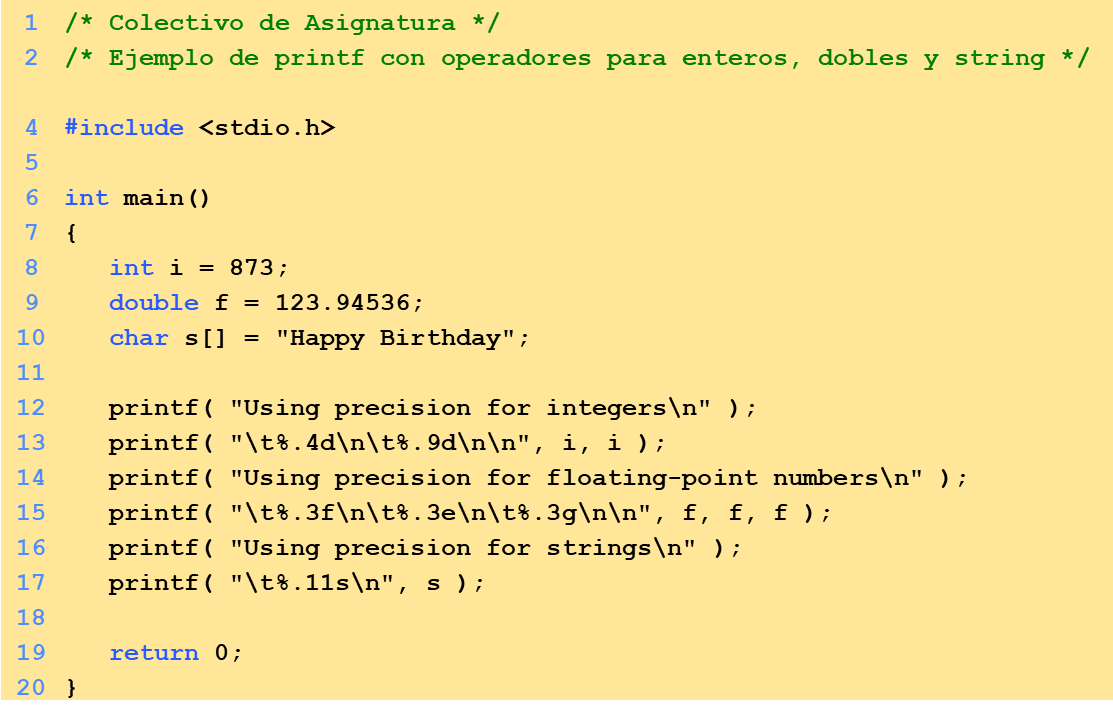
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Carácter de conversión** | **Significado** |  | **Secuencia de**  **Escape** | **Significado** |
| **c** | El dato es un carácter. |  | \b | Espacio atrás |
| **d** | El dato es un entero decimal. |  | \f | Salto de página |
| **e** | El dato es un 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. |  |

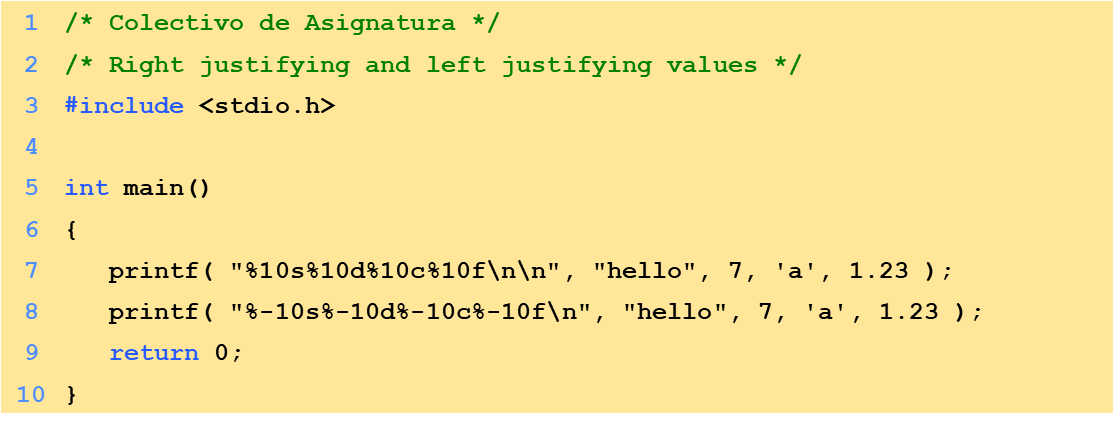
1. **Implementando funciones de librerías para E/S con formato**

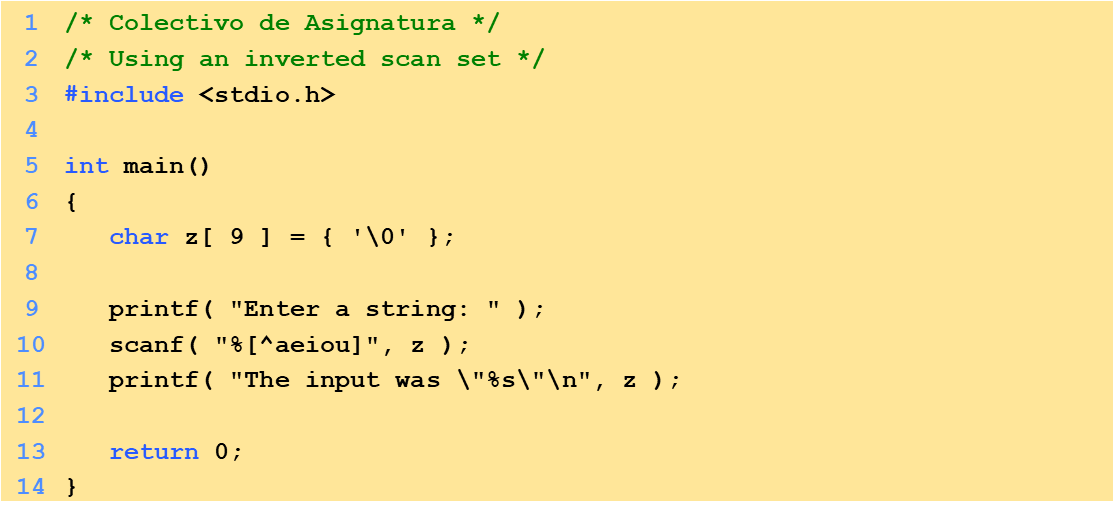
Según la imagen presentada copie el código y compile.

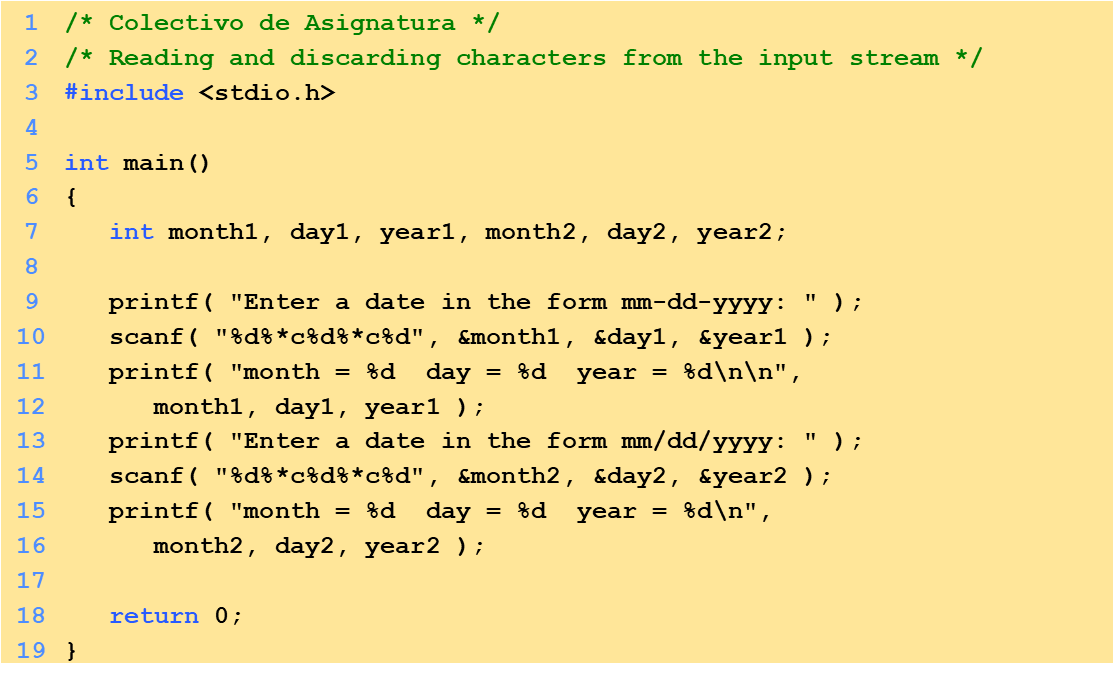
**Ejercicio#1** **Ejercicio #2**

**Ejercicio #3**



 **Ejercicio #4**

**Ejercicio #5**

 **Ejercicio #6**