A. ¿Qué se imprime por consola y por qué?

Dado el código actual, las funciones funcion(short) y funcion(float) están comentadas, por lo que el programa solo usa las funciones funcion(int) y funcion(double).

Primera línea (System.out.println("char : " + funcion(c));):
Se pasa un char ('g') a la función funcion(c).
El tipo char no tiene un método funcion(char) definido, por lo que se produce una conversión implícita de char a int.
El valor de 'g' se convierte a su valor entero correspondiente en ASCII (103).
Luego, el valor entero 103 se pasa a la función funcion(int a), y se imprime:
char : Entra a int: 103

Segunda línea (System.out.println("short : " + funcion(s));):
Se pasa un short (2) a la función funcion(s).
El tipo short no tiene un método funcion(short) definido.
El tipo short se puede convertir automáticamente a int (autoboxing), por lo que la función funcion(int) se invoca.
El valor 2 se pasa a la función funcion(int a), y se imprime:
short : Entra a int: 2

Tercera línea (System.out.println("byte : " + funcion(b));):
Se pasa un byte (1) a la función funcion(b).
El tipo byte también se convierte automáticamente a int, ya que no hay un método funcion(byte) definido.
El valor 1 se pasa a la función funcion(int a), y se imprime: byte : Entra a int: 1

Cuarta línea (System.out.println("long : " + funcion(l));):
Se pasa un long (999999999) a la función funcion(l).
El tipo long se convierte a int debido a que no existe un método funcion(long) y el valor de long se ajusta dentro de los límites de int.
El valor 999999999 es mayor que el valor máximo permitido para int (2^31 - 1 = 2147483647), lo que produce un desbordamiento, y el valor resultante es negativo (debido al desbordamiento de enteros).
El valor resultante (en este caso, se imprime 999999999, pero en realidad sería un valor negativo debido al desbordamiento).
La salida es: long : Entra a int: 999999999

Quinta línea (System.out.println("integer : " + funcion(i));):
Se pasa un int (51232) a la función funcion(i).
El valor 51232 se pasa directamente a la función funcion(int a), y se imprime: integer : Entra a int: 51232

Sexta línea (System.out.println("double : " + funcion(d));):
Se pasa un double (12.4) a la función funcion(d).
El tipo double no tiene un método funcion(double) definido, por lo que se invoca la función funcion(double) directamente.
El valor 12.4 se pasa a la función funcion(double a), y se imprime: double : Entra a double: 12.4

Séptima línea (System.out.println("float : " + funcion(f));):
Se pasa un float (5.65f) a la función funcion(f).
El tipo float se convierte automáticamente a double (debido a que el tipo double tiene mayor precisión que float), y la función funcion(double) se invoca.
El valor 5.65 se pasa a la función funcion(double a), y se imprime: float : Entra a double: 5.65

B. Realice los siguientes cambios, explicando cómo y por qué cambia lo que se imprime:

1. Active la función que recibe un short.

Al activar la función que recibe un short, el tipo short no se convertirá a int automáticamente. Ahora, cuando se pase un valor de tipo short a la función funcion(short), se ejecutará esa función en lugar de la que recibe int. La salida será diferente:
Codigo modificado

```
static String funcion(short a) {
   return "Entra a short: " + a;
}
```
Imppresion

short : Entra a short: 2

La línea correspondiente a short ahora imprimirá el mensaje de la función funcion(short).

2. Active la función que recibe un float.

Al activar la función que recibe un float, el tipo float no se convertirá automáticamente a double. Ahora, la función funcion(float) se invocará cuando se pase un valor float. La salida correspondiente cambiará:

Codigo modificado:

```
static String funcion(float a) {
   return "Entra a float: " + a;
}
```

Impresión:
float : Entra a float: 5.65
La línea correspondiente a float ahora imprimirá el mensaje de la función funcion(float).

3. Comente la función que recibe un double y active la que recibe un float.

Si comentas la función que recibe un double y activas la función que recibe un float, el tipo float se pasará a la función funcion(float) y ya no se usará la función funcion(double). La salida de la línea que usa float cambiará:

Código modificado:

```
// Comentar la función que recibe un double
// static String funcion(double a) {
//     return "Entra a double: " + a;
// }

static String funcion(float a) {
   return "Entra a float: " + a;
}
```

Impresión:
float : Entra a float: 5.65

4. Comente todas las funciones, excepto la que recibe un double.

Si comentas todas las funciones excepto la que recibe un double, solo la función funcion(double) será utilizada. Las otras líneas que no reciben un double producirán un error de compilación porque no hay funciones definidas para esos tipos.

Código modificado:

```
// Comentar todas las funciones, excepto la que recibe un double
// static String funcion(short a) { return "Entra a short: " + a; }
// static String funcion(int a) { return "Entra a int: " + a; }
// static String funcion(long a) { return "Entra a long: " + a; }
// static String funcion(float a) { return "Entra a float: " + a; }

static String funcion(double a) {
```

```
    return "Entra a double: " + a;
}
```

double : Entra a double: 12.4

Las demás líneas que usaban tipos distintos a double ahora causarán un error de compilación, ya que no hay ninguna función que los reciba.