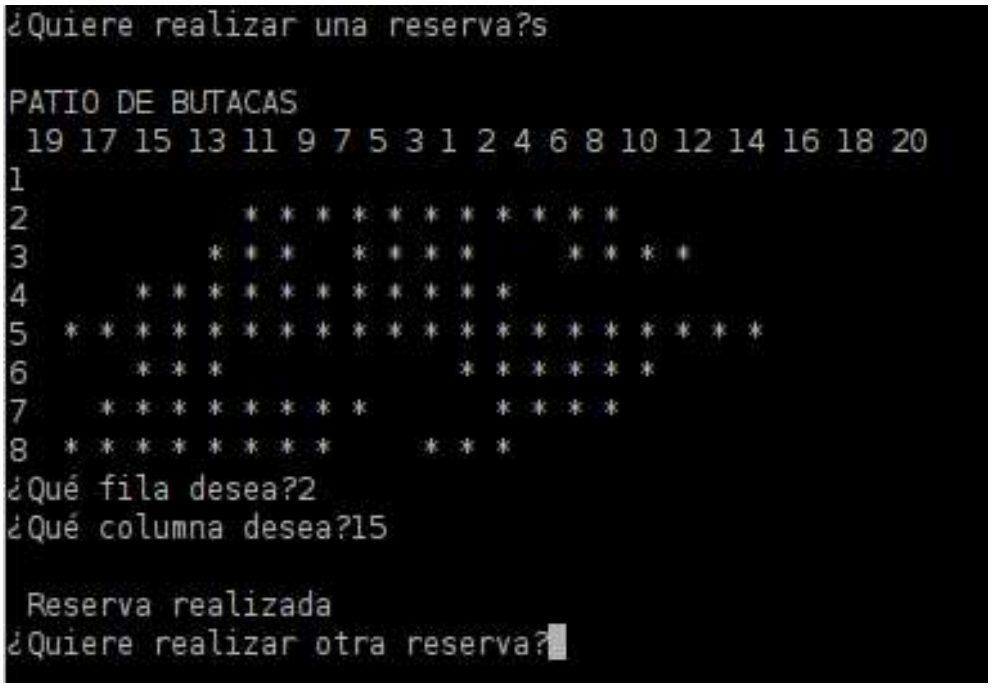
EJERCICIOS DE ARRAYS

Problema 1: El cine

Una sala de cine necesita que le diseñemos un programa en C para poder reservar los asientos de la sala. La sala se compone de 8 filas y cada una de ellas tiene 20 butacas. Del centro de la sala a la derecha encontramos las butacas pares, mientras que del centro a la izquierda encontramos los números impares.

Para realizar una reserva el programa deberá mostrar en primer lugar qué butacas están libres, y preguntar al cliente qué butacas desea. Cuando el cliente solicite una butaca, el programa deberá comprobar si esas butacas están libres, si lo están las reservará sino deberá mostrar un mensaje de error.

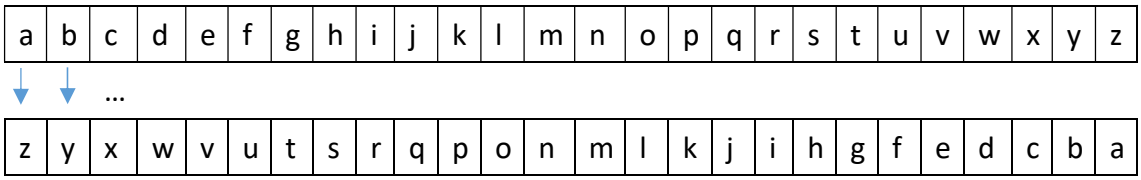
Ejemplo de ejecución:



Problema 2: Ayudando de nuevo al agente 00Negativo

El famoso agente doble Juan Bones utiliza el siguiente método para codificar sus mensajes: reemplaza cada carácter por su imagen especular. Para que no sea muy complicado, sólo usa letras minúsculas, dígitos y espacios en blanco. Las letras se transforman en su imagen especular, los dígitos y los espacios en blanco se dejan como están.

La siguiente imagen refleja cual es la imagen especular de cada carácter:



El agente 00Negativo ha averiguado el método que usa el agente Bones y nos ha encargado un programa que pueda decodificar sus mensajes. Para ello el programa pide el mensaje y a continuación lo muestra decodificado.

Hay que tener en cuenta que debido a errores en la transmisión, puede haber caracteres no permitidos en el mensaje. Sólo deben ser procesados las letras minúsculas, los dígitos y los espacios en blanco, el resto de caracteres deben ser ignorados.

Para determinar si un carácter es una letra minúscula puedes usar la función `islower()` y para determinar si un carácter es un dígito puedes emplear `isdigit()`.

Casos de prueba

Entrada	**Salida**
sloz vhgfwrzmgv wv k1	hola estudiante de p1
ri z 25 mligv 56 hfi	ir a 25 norte 56 sur
Hsr uirvmw	hi friend

¿Puedes decodificar este mensaje? “yfvmz hfvigv vm k1”

Problema 3: Escapa del laberinto

Implementa un juego en el que el jugador se mueve en un mapa bidimensional formado por una cuadrícula de habitaciones (mazmorras). Cada una de ellas puede contener un monstruo, un tesoro, la salida, o bien nada. El objetivo es escapar por la salida sin ser devorado por ningún monstruo. Las reglas son las siguientes:

- El jugador siempre partirá de la esquina superior izquierda (fila 0, columna 0)
- La casilla con la salida de las mazmorras se fijará de forma aleatoria, pero no puede ser la (0,0)
- El tamaño del mundo (filas y columnas) lo elige el jugador al principio de la partida
- Los monstruos y tesoros se repartirán de forma aleatoria. En cada casilla debe haber un 10% de probabilidades de que haya un monstruo y un 20% de que haya un tesoro.
- En el modo de juego normal, el mundo no será visible para el jugador. Únicamente se le informará del contenido de la habitación actual cuando se mueva a ella. Pero para poder probar y depurar el juego debería haber también un modo en el que el mapa sea visible.

- Los movimientos se introducirán como letras: n-norte, s-sur, e-este, o-oeste. Hay que controlar los movimientos "imposibles" cuando estamos en el borde del mapa (por ejemplo no se puede ir al norte estando en la primera fila, o al este estando en la última columna)
- Cuando se llega a una habitación con un monstruo, este devora al jugador y el juego termina inmediatamente.
- Cuando se llega a un tesoro, el jugador se lo lleva, incrementando el número de tesoros que tiene. La habitación pasará a estar vacía.
- Si el jugador alcanza la salida, el juego acaba, imprimiendo el número de tesoros que ha conseguido.

```
¿Cuántas filas quieres? 3
¿Cuántas columnas quieres? 3
¿Quieres que el mundo sea visible? (s/n) s
(J=Jugador, M=Monstruo, T=Tesoro, S=Salida)
 J  .  S
 M  .  .
 .  T  M
Introduce el movimiento (n/s/e/o): n
¡Ouch! ¡Te has dado contra una pared!. Prueba otra dirección
Introduce el movimiento (n/s/e/o): s
Un monstruo te devora. Todo ha terminado
```

(otra ejecución)

```
¿Cuántas filas quieres? 3
¿Cuántas columnas quieres? 3
 J  T  .
 S  .  .
 .  T  .
Introduce el movimiento (n/s/e/o): e
¡¡Encuentras un tesoro!!
 .  J  .
 S  .  .
 .  T  .
Introduce el movimiento (n/s/e/o): s
```

```
Estás en una mazmorra fría y oscura...

 .  .  .

 S  J  .

 .  T  .

Introduce el movimiento (n/s/e/o): o

 .  .  .

 J  .  .

 .  T  .

¡Has salido con vida de las mazmorras! Te llevas 1 tesoros!
```

Pistas de implementación:

Puedes **representar el mundo** con un array bidimensional de enteros, codificando los posibles contenidos de una habitación con diferentes valores. Por ejemplo 0 podría significar vacío, 1 un monstruo, 2 un tesoro….

Rellenar con monstruos/tesoros: debes ir recorriendo todas las filas y columnas. Para ver si la casilla actual debería contener un monstruo puedes generar un número al azar entre 0 y 100 (o sea, con la instrucción rand()%100) y comprobar si es menor que 10. Esto solo pasará el 10% de las veces. Si no, aplica el mismo procedimiento (pero con el 20%) para ver si debe contener un tesoro y finalmente si no pasa nada de esto debe estar vacía.