

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**  
**FACULTAD DE CIENCIAS E INGENIERÍA**

**ALGORITMIA**  
**1ra práctica (tipo b)**  
**(Segundo semestre 2014)**

**Indicaciones generales:**

- Duración: 2h 45 min.
- Materiales o equipos a utilizar: No se permite el uso de material de consulta.
- Al inicio de cada programa, el alumno deberá incluir, a modo de comentario, la estrategia que utilizará para resolver el problema. De no incluirse dicho comentario, el alumno perderá el derecho a reclamo en esa pregunta.
- Si la implementación es significativamente diferente a la estrategia indicada, la pregunta será corregida sobre el 50 % del puntaje asignado y sin derecho a reclamo.
- Un programa que no muestre resultados coherentes y/o útiles será corregido sobre el 60 % del puntaje asignado a dicha pregunta.
- Debe utilizar comentarios para explicar la lógica seguida en el programa elaborado.
- Cada programa debe ser guardado en un archivo con el nombre *preg#\_<codigo\_de\_alumno>.c* y subido a PAIDEIA en el espacio indicado por los Jefes de Práctica.

Puntaje total: 20 puntos

---

**Cuestionario:**

**Pregunta 1 (4 puntos) Reconociendo *h-sequences***

Una *h-sequence* está formada por un único 0 o por un 1 seguido de dos *h-sequences*. Una gramática de *h-sequences* podría ser la siguiente:

$$\langle h \rangle = 0|1\langle h \rangle\langle h \rangle$$

<b>SON <i>h-sequences</i></b>	<b>NO SON <i>h-sequences</i></b>
0	1
100	1001
10100	11010
110010100	1001110
110100100	1010101

Implemente una función **recursiva** en ANSI C que reciba como parámetro un arreglo de  $N$  elementos ( $N > 0$ ) y determine si es una *h-sequence* o no. Puede asumir que el arreglo solo contiene 0s y 1s.

## Pregunta 2 (7 puntos) UVa 10662 - The Wedding (Traducción Libre)

Félix y Leti se van a casar pronto y todos están preparando sus regalos de boda, pero tienen un serio problema: el dinero; su presupuesto no es muy alto. A ellos les gustaría reservar un buen restaurante, “dormir” su primera noche en un buen hotel y pasar una maravillosa luna de miel viajando alrededor del mundo. La mejor manera de obtener el precio más barato es adquirir un paquete todo incluido; es decir, se tiene que contratar el viaje, el restaurante y el hotel todo junto. ¿Es eso posible?

El problema consiste en encontrar la combinación de agencia de viaje - restaurante - hotel más barata. La “dificultad” radica en que no todas las combinaciones son posibles.

**Entrada:** Cada caso de prueba tiene el siguiente formato:

- La primera línea consiste de 3 números enteros  $T$ ,  $R$ ,  $H$  que indican el número de agencias de viaje, restaurantes y hoteles, respectivamente. Asuma que  $T < 20$ ,  $R < 20$  y  $H < 20$ . Las agencias de viaje, restaurantes y hoteles están numerados: 0, 1, 2, ...
- Las siguientes  $T + R + H$  líneas están divididas en tres bloques:
  - El primer bloque tiene  $T$  filas y  $R + 1$  columnas. La primera columna corresponde a los precios ofrecidos por las agencias de viajes para el viaje alrededor del mundo. En el resto de columnas, la celda  $(i, j)$  es 0 si la agencia de viajes  $i$  se puede combinar con el restaurante  $j$  y 1 si no es posible.
  - El segundo bloque tiene  $R$  filas y  $H + 1$  columnas. La primera columna corresponde a los precios ofrecidos por los restaurantes. En el resto de columnas, la celda  $(i, j)$  es 0 si el restaurante  $i$  se puede combinar con el hotel  $j$  y 1 si no es posible.
  - El tercer bloque tiene  $H$  filas y  $T + 1$  columnas. La primera columna corresponde a los precios ofrecidos por los hoteles. En el resto de columnas, la celda  $(i, j)$  es 0 si el hotel  $i$  se puede combinar con la agencia de viajes  $j$  y 1 si no es posible.
- La entrada culmina con una línea en blanco.

**Salida:** Para cada caso de prueba, la salida debe consistir de una sola línea con el número de agencia de viajes ( $T$ ), restaurante ( $R$ ) y hotel ( $H$ ), y el precio total más barato ( $P$ ). Estos valores deben ser escritos en el formato: `T R H: P`. Si no hay combinación posible, la salida debe ser una línea con el texto: `Don't get married!`. Si existe más de una posibilidad, puede escribir cualquiera.

Ejemplo de Entrada:	Ejemplo de Salida:
2 2 2 12 0 0 1 1 1 34 0 0 3 1 1 21 1 0 2 1 0 2 2 2 12 0 0 1 0 0 34 0 0 3 0 0 21 0 0 2 0 0	Don't get married! 1 1 1:6

## Pregunta 3 (9 puntos) D-path

Se dispone de un tablero de caracteres de 15x15, en cuyos casilleros se pueden encontrar una 'X', un espacio en blanco ( ' ' ) o una 'D' tal como se muestra en la siguiente figura:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	X							X			X	X	X		X
1			X				X	X				X			
2							X								
3	X			X	X	X						X			X
4											X	X	X	X	
5							X								X
6				X	X	D									X
7					X	X								X	
8					X		X						X		
9		X		X	X			X					X		
10					X					X	X	X			
11					X							X	X		
12			X			X				X					
13	X	X				X									
14					X	X									

Como se puede apreciar, la letra 'D' se encuentra rodeada de algunas letras 'X' y estas también pueden estar dispersas en otros casilleros del tablero.

Se desea elaborar un programa que permita determinar una ruta de desplazamiento de la letra 'D' (única en el tablero). La siguiente figura muestra la ruta que finalmente seguirá la letra 'D' :

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	X							X			X	X	X		X
1			X				X	X				X			
2							X								
3	X			X	X	X						X			X
4											X	X	X	X	
5							X								
6				X	X	D									
7					X	X								X	
8					X										
9		X		X	X										
10					X					X	X	X			
11					X							X	X		
12			X			X				X					
13	X	X				X									
14					X	X									

Para determinar esta ruta, el programa deberá seguir los siguientes pasos:

- Determinar la dirección en la cual exista la **mayor** cantidad de 'X' adyacentes a una 'D' considerando, además, que al extremo de las 'X' debe haber un casillero en blanco. Para este paso **deberá usar, por lo menos, una función recursiva**. En el tablero que se da como ejemplo, la dirección será **Diagonal Inferior Derecha**. Observe que en la dirección **Abajo** hay más 'X' pero no se puede considerar esta dirección ya que no hay un casillero vacío al extremo.
- Colocar la letra 'D' en el casillero en blanco y reemplazar todas las 'X' que se encuentran en la dirección seleccionada por '+'. En el ejemplo anterior, se coloca la letra 'D' en el casillero (11, 10), reemplazando por '+' las 'X' que se encuentran en los casilleros (7, 6), (8, 7), (9, 8) y (10, 9).
- Repetir los pasos anteriores hasta que no haya posibilidad de mover la letra 'D'.

Al final, el tablero quedaría como se muestra en la siguiente figura:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	X							X			X	X	X		X
1			X				X	X				X			
2							X								
3	X			X	X	X						X			X
4											D	+	+	+	D
5							X								+
6				X	X	D									+
7					X	+								X	D
8					X			+						+	
9		X		X	X				+				+		
10					X					+	X	+			
11					X						D	X	X		
12			X			X				X					
13	X	X				X									
14					X	X									

Implemente un programa en ANSI C que resuelva el problema descrito. Para ello, el tablero se leerá desde el archivo de texto *dpath.in* con redireccionamiento de la entrada estándar. En la primera línea de dicho archivo se encuentran las coordenadas de la letra 'D' en el tablero y luego, en las siguientes líneas, aparecen los caracteres que corresponden al tablero, siguiendo la distribución correspondiente de filas y columnas. Luego de encontrar el camino siguiendo el algoritmo planteado, deberá imprimir el tablero final en el archivo *dpath.out* con redireccionamiento de la salida estándar.

Profesor del curso: Fernando Alva

San Miguel, 02 de setiembre de 2014