Universidad Simón Bolívar
Departamento de Computación y Tecnologías de la Información
CI2126 Computación II

Práctica 2: Recursión

1) Escriba una función recursiva en C que invierta una cadena de caracteres.
2) Sea ***p*** un arreglo de caracteres, p es un palíndrome si se cumple que al invertirlo se obtiene la misma secuencia de caracteres. Escriba la definición recursiva y una función en lenguaje C que retorne TRUE si ***p*** es un palídrome y FALSE en caso contrario.
3) Dados dos strings S1 y S2 escriba una función recursiva que verifique si S2 se encuentra en S1. En caso de encontrarse devuelva la posición dentro del string S1 donde comienza S2 en caso contrario devuelva el mensaje "No encontrado".
4) Escriba una función iterativa y otra recursiva que convierta un número decimal a binario.
5) Escriba una función iterativa y otra recursiva en C que multiplique dos matrices *(a*b)*, donde *a* es de tamaño (n*m) y *b* es de tamaño (o*p).
6) Escriba una función recursiva para calcular el Máximo Común Divisor (MCD) de dos números n1 y n2, utilizando la siguiente definición :
   - MCD(n1, n2)=n2 si (n2<=n1 && n1 % n2==0)
   - MCD(n1, n2)= MCD(n2, n1) si (n1<n2)
   - MCD(n1, n2)= MCD(n2, n1%n2) en otros casos.
7) Escriba un programa recursivo para calcular la determinante de una matriz (n*n).
8) Se define el punto de ensilladura de una matriz de tamaño mxn como la posición i,j que contiene al mínimo valor de la fila i que corresponde al máximo valor de la columna j. Por ejemplo, si la matriz es la siguiente (mxn=3x2):

1	5
14	9
8	6

La misma posee un punto de ensilladura en la posición 2,2 (mínimo valor de la fila 2 que corresponde al máximo valor de la columna 2). Escriba una función en C que escriba los valores de i y de j del punto de ensilladura de una matriz dada en el caso de existir. En caso contrario debe devolver los valores -1,-1.

9) Elaborar un programa recursivo que reciba la siguiente matriz que simbolice un laberinto, de manera que una casilla con valor L identifique a un Ladrillo (no pase) y una casilla con el caracter E identifique un espacio o un camino libre (si paso).

```
lab[FILAS][COLUMNAS] =
   { { L, L, L, L, L, L, L, L, L, L, L, L, L, L, L, L, L, L, L, L },
     { E, E, E, L, E, L, E, L, E, L, E, E, E, L, E, E, E, E, E, L },
     { L, L, E, L, E, E, E, L, E, E, E, L, E, L, E, L, E, E, L, E, L },
     { L, E, E, E, E, L, E, L, L, L, E, L, E, L, E, L, L, E, L, E, L },
     { L, E, L, L, L, L, E, L, E, L, E, L, E, L, L, L, E, E, L, E, L },
     { L, E, E, E, L, E, E, L, E, L, E, L, E, E, E, E, E, L, L, E, L },
     { L, E, L, L, L, L, L, L, E, L, E, L, L, E, L, E, E, E, E, E, L },
     { L, E, E, E, E, E, E, E, E, E, E, E, L, E, L, E, L, L, E, L, L },
     { L, L, L, L, L, L, L, L, L, L, L, E, L, E, L, E, E, L, E, L, L },
     { L, E, E, E, E, E, E, E, L, E, E, E, L, L, L, L, L, L, E, L, L },
     { L, E, L, L, L, L, L, E, E, E, L, L, L, E, L, E, E, E, E, E, L },
     { L, E, E, L, E, E, E, E, L, E, E, L, E, E, E, E, L, L, L, E, E },
     { L, L, L, L, L, L, L, L, L, L, L, L, L, L, L, L, L, L, L, L, L }
   };
```

Además debe recibir una posición (i,j) que refleja la posición inicial dentro de la matriz. El programa debe indicar si el laberinto, representado por la matriz, tiene salida. La idea es hacer un programa que simule el recorrido dentro del laberinto a través de los caminos libres.