

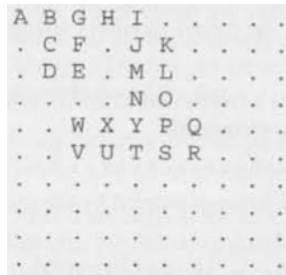
Programación y Algoritmos

Parte 1: Lenguaje C

Tarea #3

Fecha de entrega: 9/10/2021.

1. Escribir un programa que genere una caminata aleatoria en una matriz de 10x10. El arreglo debe contener inicialmente puntos '.', y debe recorrerse basado en el residuo de un número aleatoria (usar srand y rand()) cuyos resultado puede ser {0 (arriba), 1 (abajo), 2 (izq), 3 (der)}, que indican la dirección a moverse. A) Verificar que el movimiento no se salga del arreglo de la matriz, y B) No se puede visitar el mismo lugar más de una vez. Si alguna de estas condiciones intentar moverse hacia otra dirección; si todas las posiciones están ocupadas, finalizar el programa e imprimir el resultado. Ejemplo:



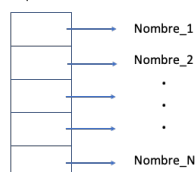
'Y' esta bloqueado por todos los lados, por lo que no puede continuar con la 'Z'. Puede hacer uso de letras o números en orden creciente para mostrar los lugares visitados. Hacer uso de funciones cuyo argumento incluya el arreglo 2D; esto es, una función para inicializar el arreglo, otra para realizar la caminata aleatoria, y otra para escribir la caminata.

2. Dado un archivo de entrada, escribir un programa que encuentre los siguiente:
 - a) Probabilidad (Pbb) de aparición de cada una de las letras del alfabeto (no haga diferencia entre minúsculas y mayúsculas).
 - b) $Pbb(x|y)$, para $(x,y) \in (a,...,z,A,...,Z)$ de las 10 letras (x) más frecuentes.

El archivo debe ser recorrido solo una vez (se evaluará la eficiencia de su código).

3. Dado una lista de nombres (strings) de N personas (apellido_paterno, apellido_materno, Nombre(s)), escribir un programa que ordene los nombres alfabéticamente usando un arreglo de apuntadores:

Arreglo de apuntadores



Los nombres pueden tener distintas longitudes; cuando un nombre sea prefijo de otro, considerar al nombre mas corto como menor. El ordenamiento debe ser a través de una función que reciba el arreglo de punteros.

4. Programa que encuentre los tres números mayores de un arreglo de enteros, especificando su posición (índice) original en el arreglo de entrada. El prototipo de la función debe ser:

```
void find_three_largest(const int *a, int len_a, int *largest, int *index_largest,
                        int *second_largest, int *second_index_largest,
                        int *third_largest, int *third_index_largest);
```