Tarea 4 - Programación y algoritmos
Giovanni Gamaliel López Padilla

Problema 1

Escribir un programa que genere una caminata aleatoria en una matri de 10x10. El arreglo debe contener inicialmente puntos ".", y debe recorrers 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 pued 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 imprimi el resultado.

Para este problema se definidio como una constante global el tamaño de la matriz y el maximo de letras posibles. Los valores son 10 y 26 respectivamente. Esto porque existen 26 letras en el código ascii ignorando a la letra ñ. El estado inicial de la matriz contiene unicamente puntos, esto puede verse en la figura 1. El programa inicia de dos maneras diferentes, un usuario introduce un punto inicial donde comenzará la caminata o este es escogido aleatoriamente.

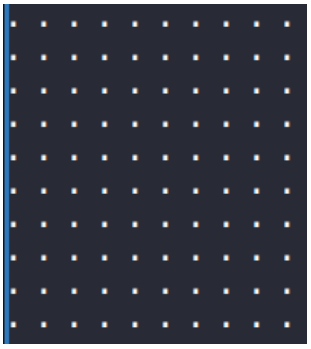

Figura 1: Estado inicial de la matriz

Una casilla se considera disponible si su valor es igual a ".". Antes de obtener el siguiente paso de cada iteracion el programa verifica si sus vecinos estan disponibles. Si existe al menos una casilla disponible entonces el programa sigue (figura 2), si no el programa termina ya que no podra seguir la caminata (figura 3). El siguinte paso de la iteración se obtiene con la función *rand()*. Se calcula el modulo del número obtenido con 4. Con esto nos aseguramos que el número se encuentre en el conjunto $\{0, 1, 2, 3\}$.

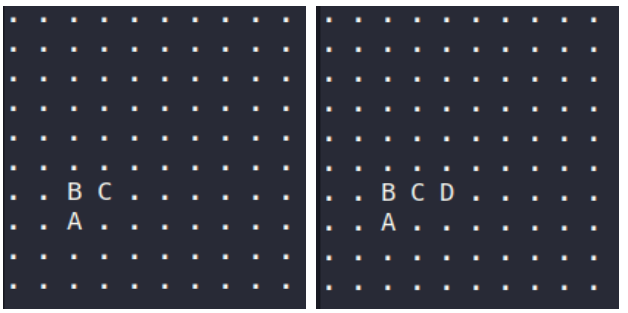

Figura 2: Ejemplo de caminata.

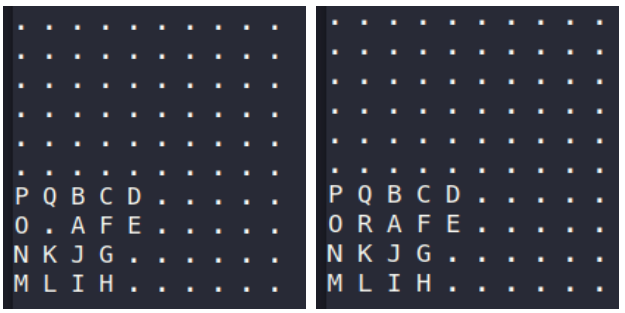

Figura 3: Ejemplo de termino del programa.

Cada paso debe estar contenido dentro del arreglo de 10x10, si el paso sobrepasa la barrera este es rechazado y se calcula otro diferente. Esto es realizado con la función is_in_the_box con el siguiente algoritmo:

```
int is_in_the_box(int pos[])
{
for (int i = 0; i < 2; i++)
{
    if ((pos[i] < 0) || (pos[i] >= size))
    {
        return 0;
    }
}
return 1;
}
```

Si esta dentro de los límites la función devuelve 1, si esta fuera 0. En la figura 4 se muestran diferentes resultados obtenidos por el programa:

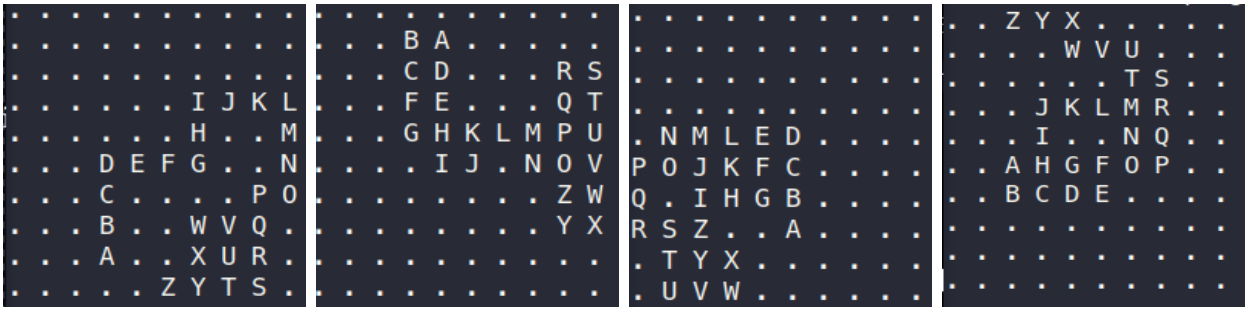

Figura 4: Diferentes output del programa

El programa se encuentra en la carpeta Problema_1. Para compilar el programa se uso el siguiente comando:

```
gcc -Wall -Wextra -Werror -pedantic -ansi -o main.out main.c -std=c11
```

Problema 2

Dado un archivo de entrada, escribir un programa que encuentre los siguiente:

Problema 2a

Probabilidad (P) de aparición de cada una de las letras del alfabeto (no haga diferencia entre minúsculas y mayúsculas)

Problema 2b

P(x|y), para (x,y) $\in$ (a,...,z,A,...,Z) de las 10 letras (x) más frecuentes.

Problema 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. 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 apuntadores.

Problema 4

Programa que encuentre los tres números mayores de un arreglo de enteros, especificando su posición (índice) original en el arreglo de entrada.