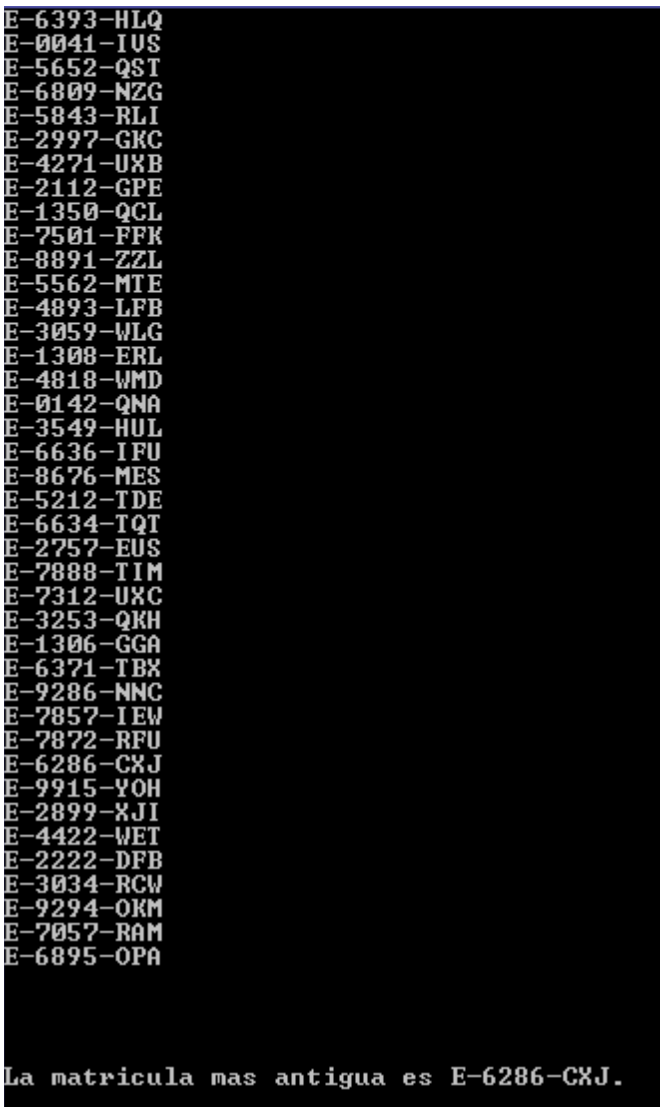
fEn esta práctica se pide la implementación de un conjunto de funciones y un programa principal para el manejo de cadenas de caracteres junto con matrices. Las cadenas van a ser matrículas de tráfico con el formato E-DDDD-LLL, donde D es un dígito y L una letra del alfabeto (sin incluir la Ñ), siendo válidas todas las letras aunque en la realidad haya algunas letras que no se usen. Por ejemplo, E-2166-BRB o E-8882-FYW serían matrículas válidas. Las funciones que se pide implementar son:

- **rellenar** una matriz de 40 matrículas válidas usando la función aleatorio de las prácticas anteriores.
- **imprimir** el contenido de la matriz.
- **comparar** dos matrículas. Esta función devolverá un ***valor negativo*** si la primera matrícula es más antigua que la segunda, un ***valor positivo*** si la primera matrícula es más reciente que la segunda y ***0*** si son iguales.
- **mas_antigua,** que devuelve la matrícula más antigua de las almacenadas en la matriz.

El programa principal deberá rellenar la matriz con las 40 matrículas, imprimir la matriz y también la matrícula más antigua.

Una posible ejecución de la misma es la siguiente:



SOLUCIÓN:

// includes y defines

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

#define N 11
#define M 40
```

// prototipos

```c
int aleatorio(int inf, int sup);
void rellenar(char mat[M][N]);
void imprimir (char mat[M][N]);
int comparar(char *mat1, char *mat2);
void masAntigua(char mat[M][N]);
```

// main

```c
int main(){

    char matriculas[M][N];
    srand(time(NULL));

    rellenar(matriculas);
    imprimir(matriculas);
    printf("\n\n\n");
    masAntigua(matriculas);


}
```

// definición de funciones

```c

int aleatorio(int inf, int sup){
    int aux = inf;
    if (inf > sup){
        inf = sup;
        sup = aux;
    }
    return (rand()%(sup-inf+1)+inf);
}

void rellenar(char mat[M][N]){
    int i = 0;
```

```
    int j = 0;

    for(i = 0; i < M; i++){
        mat[i][0] = 'E';
        mat[i][1] = '-';
        mat[i][6] = '-';
        mat[i][10] = '\0';
        for(j = 0; j < N; j++){
            if(j > 1 && j < 6){
                mat[i][j] = aleatorio('0', '9');
            }
            else if(j > 6 && j < 10){
                mat[i][j] = aleatorio('A', 'Z');
            }
        }
    }
}

void imprimir(char mat[M][N]){
    int i = 0;
    int j = 0;

    for(i = 0; i < M; i++){
        for(j = 0; j < N; j++){
            printf("%c", mat[i][j]);
        }
        printf("\n");
    }
}

int comparar(char *matricula_1, char *matricula_2){

    int comparation = 0;

    comparation = strcmp(matricula_1 + 7, matricula_2 + 7);
    if(comparation == 0){
        comparation = strncmp(matricula_1 + 2, matricula_2 + 2, 4);
    }
    return comparation;
}

void masAntigua(char matriz[M][N]){

    int i = 0;
    char antiguo[M + 1];
    char matricula[M + 1];

    for(i = 0; i < M; i++){
        matricula[i] = 0;
    }

    strncpy(antiguo,matriz[0], M);
    for(i = 1; i < N; i++){
```

```
        strncpy(matricula, matriz[i], M);
        if(comparar(matricula, antiguo) == -1){
            strncpy(antiguo, matricula, M);
        }
    }

    pr
```

...