En esta práctica se pueden usar los **elementos de C permitidos** en prácticas anteriores, y hay que seguir teniendo en cuenta las **restricciones** de funcionamiento de esas prácticas.

Esta práctica está diseñada para continuar ejercitando el uso de **strings**.

Se deben **usar todos y cada uno de los prototipos** indicados en esta práctica, y los de prácticas anteriores que se requieran.

No se utilizará **asignación dinámica de memoria** en ninguna función.

En esta práctica, se deben generar aleatoriamente los DNIs, calculándose la letra, y **ordenar los DNIs ascendentemente,** printando en todo momento lo que se vaya obteniendo. Se usará el algoritmo “bubble sort”.

En la comparación de los strings se usará **memcmp()**, no strcmp().

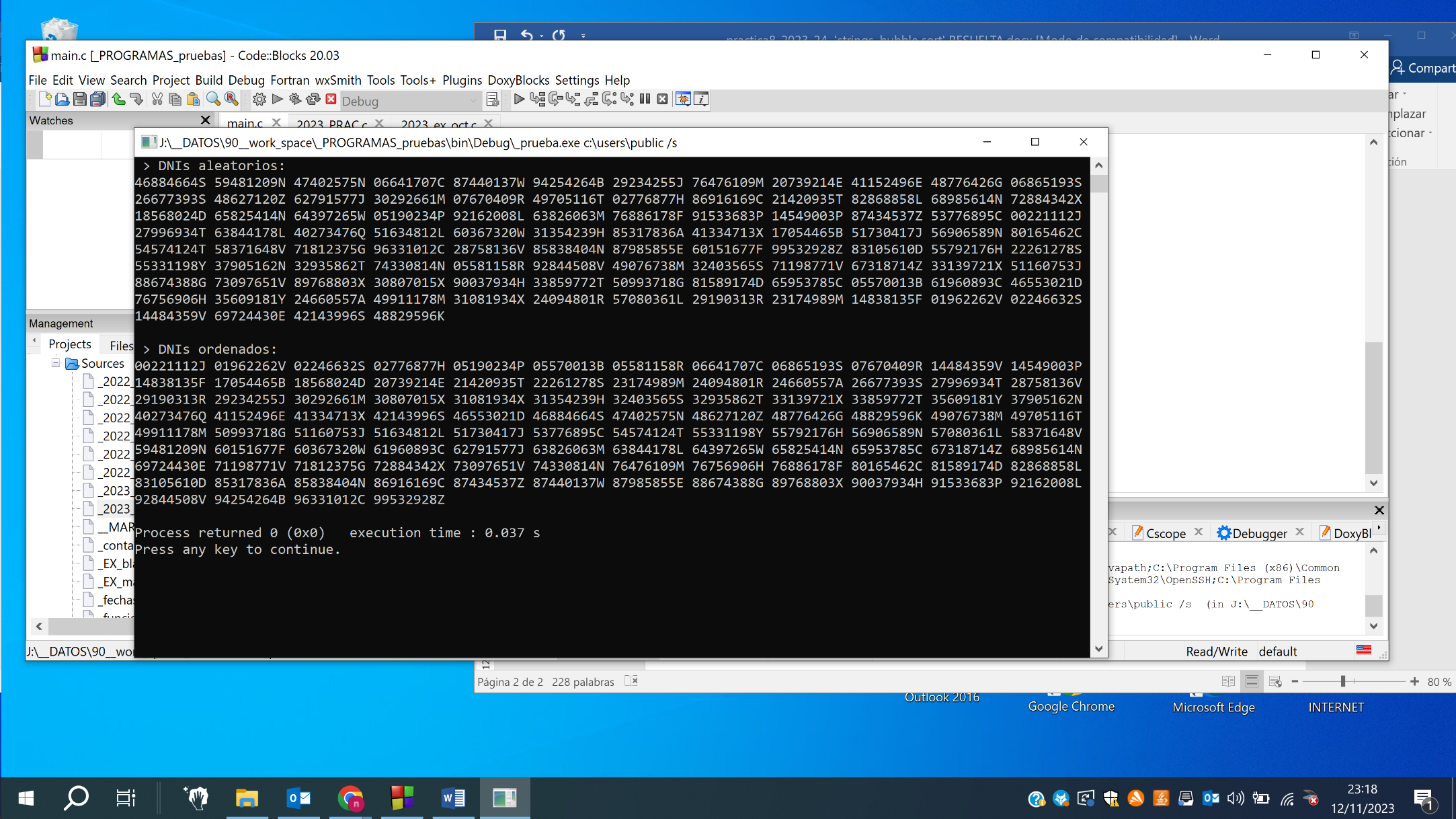


Figura 1. Ejemplo de ejecución del programa

//includes

#include ”stdio.h”

#include ”time.h” // time()

#include ”stdlib.h” // srand(), rand()

#include <string.h>

//defines

#define N 100

//prototipos usados en practicas anteriores

void print\_strings\_DNIs(char [N][10]);

void rand\_strings\_DNIs(char [N][10]);

void rand\_str\_DNI(char [10]);

unsigned resto\_DNI(unsigned );

char letra\_calculada(unsigned );

//prototipos de esta práctica

void strings\_swap(char [9+1], char [9+1]);

void strings\_bubbleSort(char [N][9+1]);

//main

int main()

{ char matriz\_DNIs[N][10];

srand(time(NULL));

rand\_strings\_DNIs(matriz\_DNIs);

printf("> DNIs aleatorios:\n");

print\_strings\_DNIs(matriz\_DNIs);

printf("\n> DNIs ordenados:\n");

strings\_bubbleSort(matriz\_DNIs);

print\_strings\_DNIs(matriz\_DNIs);

return 0;

}

// definición de las funciones

void rand\_str\_DNI(char dni[10]){

int i, num;

for(i=0;i<8;i++) //Obtenemos el Dni aleatorio en un string

dni[i]=rand()%10 + '0';

for(i=0, num = 0;i<8;i++) //Pasamos el dni en string a un tipo int para calcular la letra

num= (num\*10) + (dni[i] - '0');

dni[i] = letra\_calculada(resto\_DNI(num)); //Calculamos la letra

dni[i+1] = '\0';

}

unsigned resto\_DNI(unsigned num){

return (num%23);

}

char letra\_calculada(unsigned resto){

char letra[23] = {'T','R','W','A','G','M','Y','F','P','D','X','B','N','J','Z','S','Q','V','H','L','C','K','E'};

return letra[resto];

}

void rand\_strings\_DNIs(char matriz\_DNIs[N][10]){

int i;

char dni[10];

for(i=0;i<N;i++){

rand\_str\_DNI(dni);

strcpy(matriz\_DNIs[i],dni);

}

}

void print\_strings\_DNIs(char matriz\_DNIs[N][10]){

int i;

for(i=0;i<N;i++)

printf("%s ", matriz\_DNIs[i]);

printf("\n");

}

void strings\_bubbleSort(char matriz\_DNIs[N][10]){

int i,j;

for(i=0;i<N;i++){

for(j=N-1;j>i;j--){

if(memcmp(matriz\_DNIs[j],matriz\_DNIs[j-1],9)<0)

strings\_swap(matriz\_DNIs[j],matriz\_DNIs[j-1]);

}

}

}

void strings\_swap(char string1[10],char string2[10]){

char aux[10];

strcpy(aux,string1);

strcpy(string1,string2);

strcpy(string2,aux);

}