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 seguir ejercitando el uso de **strings** (cadenas de caracteres).

Se deben **usar todos los prototipos** indicados y sólo esos prototipos, aparte de alguno que se haya usado en prácticas anteriores.

En la práctica actual, se deben generar aleatoriamente los DNIs, calculándose la letra correspondiente, pero usando strings. Los prototipos de las funciones a implementar son los siguientes:

void buscar\_substring(char [N][9+1]);

void buscar\_letra(char [N][9+1]);

void print\_menu ();

void print\_strings\_DNIs(char [N][9+1]);

void rand\_strings\_DNIs(char [N][9+1]);

void rand\_str\_DNI(char [9+1]);

Se usará getch() para las opciones del menú. En esta ocasión, no se utilizará asignación dinámica de memoria (malloc, free).

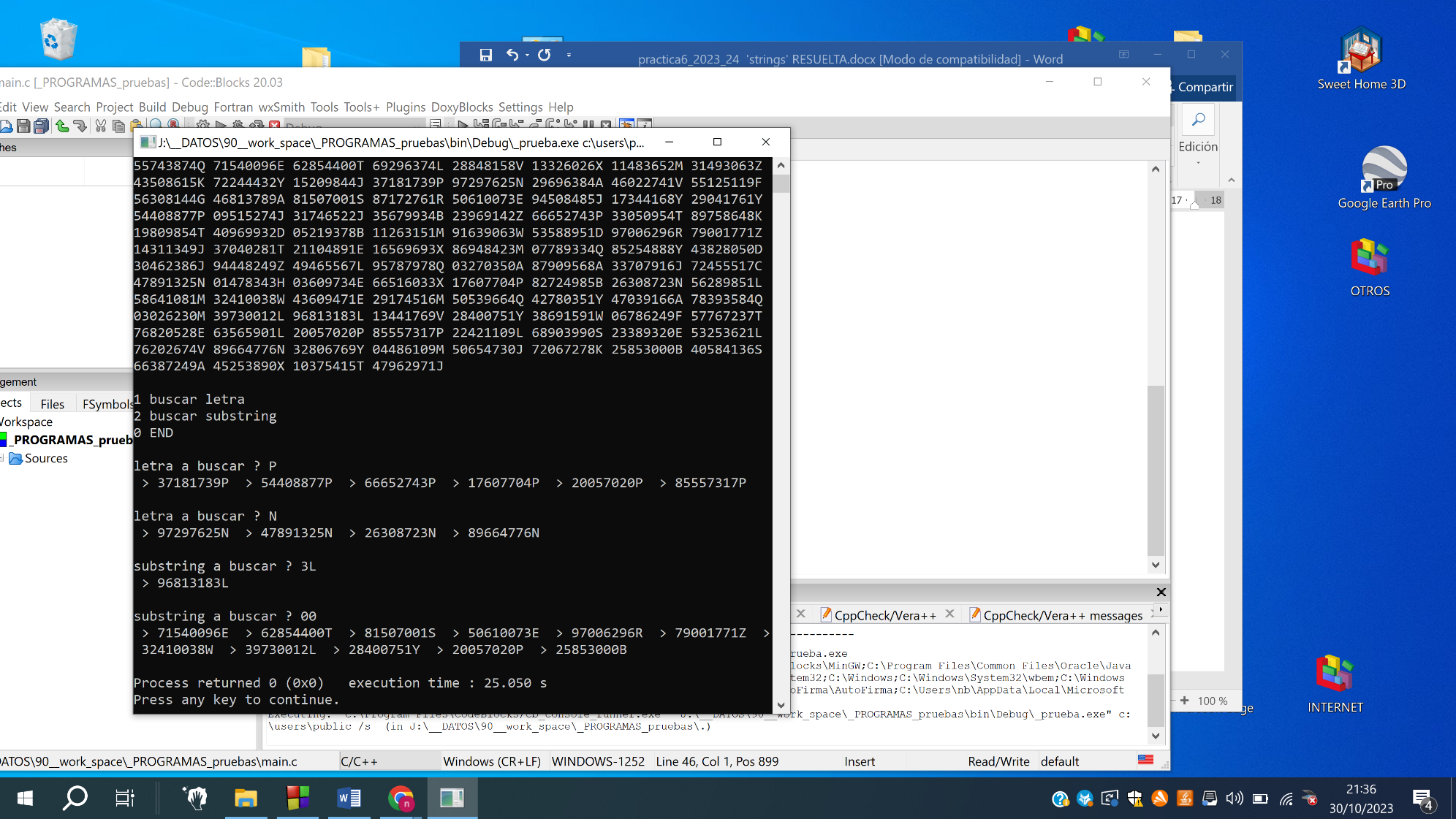


Figura 1. Ejemplo de ejecución del programa

//includes

//defines

//prototipos usados en practicas anteriores

unsigned resto\_DNI(unsigned);

char letra\_calculada(unsigned );

//prototipos de esta práctica

void buscar\_substring(char \_string\_DNIs[N][9+1]);

void buscar\_letra(char \_string\_DNIs[N][9+1]);

void print\_menu ();

void print\_strings\_DNIs(char string\_DNIs[N][9+1]);

void rand\_strings\_DNIs(char string\_DNIs[N][9+1]);

void rand\_str\_DNI(char str\_DNI[9+1]);

//main

int main()

{

char matriz\_DNIs[N][10],opcion;

srand(time(NULL));

rand\_strings\_DNIs(matriz\_DNIs);

print\_strings\_DNIs(matriz\_DNIs);

print\_menu();

do{

fflush(stdin);

opcion = getch();

if(opcion == '1')

buscar\_letra(matriz\_DNIs);

else if(opcion == '2')

buscar\_substring(matriz\_DNIs);

} while (opcion != '0');

return 0;

}

// definiciones de las funciones

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

int i, num;

for(i=0;i<8;i++) //Genera la parte numerica random

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

for(i=0, num = 0;i<8;i++) //Pasa la string a un int

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

DNI[i]=letra\_calculada(resto\_DNI(num));

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 print\_menu(){

printf("\n1 buscar letra\n");

printf("2 buscar substring\n");

printf("0 para acabar(END)\n");

}

void buscar\_letra(char matriz\_DNIs[N][10]){

char letra;

int i;

printf("Letra a buscar?: \n");

do{

fflush(stdin);

letra = getch();

if (letra >= 'a' && letra <= 'z')

letra-=32;

} while(letra < 'A' || letra > 'Z');

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

if(strchr(matriz\_DNIs[i], letra)){

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

}

}

printf("\n");

}

void buscar\_substring(char matriz\_DNIs[N][10]){

int i;

char substring[10];

printf("Substring a buscar?: \n");

fflush(stdin);

scanf("%s", substring);

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

if(strstr(matriz\_DNIs[i], substring)){

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

}

}

printf("\n");

}