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 **arrays**.

**De momento, no está permitido** el uso de strings, u otros elementos del lenguaje C.

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

En la práctica actual, se debe generar aleatoriamente los DNIs, calculándose la letra, y a continuación printarlos. A continuación, se debe **ordenar los DNIs ascendentemente** y printar el resultado de la ordenación. Se usará el algoritmo “bubble sort”, ineficiente pero didáctico. Se **utilizará** **asignación dinámica de memoria** en la función *main* y en las funciones *swap*.

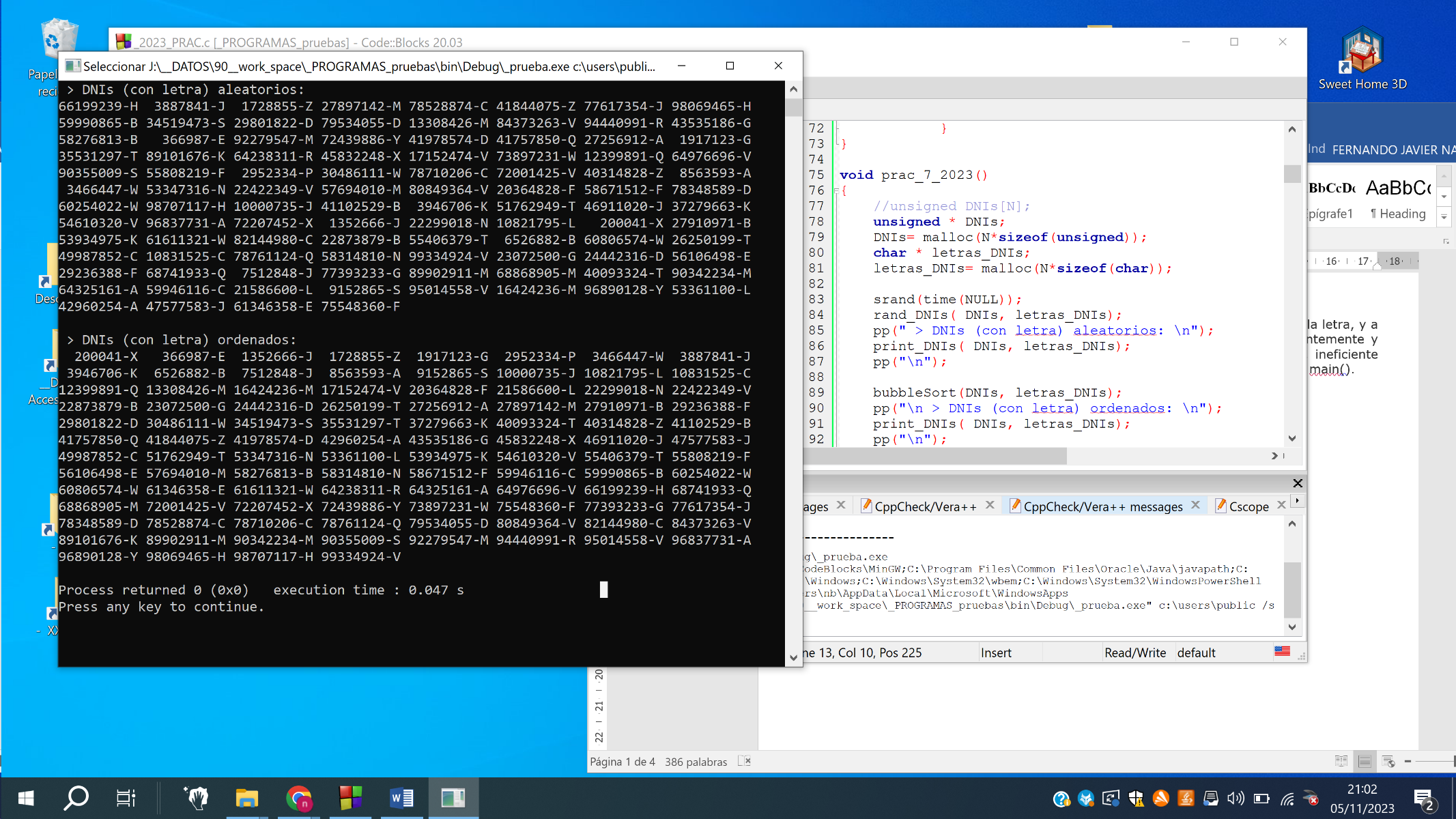


Figura 1. Ejemplo de ejecución del programa

El algoritmo **bubble sort** se puede visualizar en https://www.google.com/search?q=bubble+sort&sca\_esv=579651652&rlz=1C1RXQR\_esES974ES1028&tbm=vid&lr=lang\_es&sa=X&ved=2ahUKEwiF36veyq2CAxWxRqQEHeJoA8kQuAF6BAgaEAI&biw=919&bih=506&dpr=2.77#fpstate=ive&vld=cid:7f1bafe4,vid:pqZ04TT15PQ,st:0

//includes

//defines

//prototipos usados en practicas anteriores

void rand\_dig (char \*);

void rand\_DNI (unsigned \*);

void rand\_DNIs (unsigned [N], char [N]);

void print\_DNIs (unsigned [N], char [N]);

unsigned resto\_DNI(unsigned);

char letra\_calculada(unsigned );

//prototipos de esta práctica

void swap\_unsigned(unsigned\*, unsigned\*);

void swap\_char(char\*, char\*);

void bubbleSort(unsigned [N], char [N]);

//main

int main()

{ unsigned \*DNIs = malloc(N\*sizeof(unsigned));

char \*letras = malloc(N\*sizeof(char));

\*DNIs;

srand(time(NULL));

rand\_DNIs(DNIs,letras);

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

print\_DNIs(DNIs,letras);

bubbleSort(DNIs, letras);

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

print\_DNIs(DNIs,letras);

free(DNIs);

free(letras);

return 0;

}

// definición de las funciones

void rand\_dig (char \*digito){

\*digito=rand()%10+'0';

}

void rand\_DNI (unsigned \*dni){

int i;

char digito;

for(i=1;i<=8;i++){

rand\_dig(&digito);

\*dni = (\*dni\*10) + (digito - '0');

}

}

void rand\_DNIs (unsigned DNIs[N], char letras[N]){

unsigned dni=0, i, resto;

char letra;

for(i=0;i<N;DNIs[i]=dni, letras[i]= letra, i++, dni=0){

rand\_DNI(&dni);

resto = resto\_DNI(dni);

letra=letra\_calculada(resto);

}

}

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 print\_DNIs (unsigned dnis[N], char letras[N]){

int i;

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

printf("%08u-%c ", dnis[i],letras[i]); //El 08u muestra tambien el 0

}

void bubbleSort(unsigned DNIs[N], char letras[N]){

int i,j, aux;

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

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

if(DNIs[j]<DNIs[j-1]){

swap\_unsigned(&DNIs[j],&DNIs[j-1]);

swap\_char(&letras[j],&letras[j-1]);

}

}

}

}

void swap\_unsigned(unsigned \*menor, unsigned \*mayor){

unsigned \*aux = malloc(sizeof(unsigned));

\*aux = \*menor;

\*menor = \*mayor;

\*mayor = \*aux;

free(aux);

}

void swap\_char(char \*menor, char \*mayor){

char \*aux = malloc(sizeof(char));

\*aux = \*menor;

\*menor = \*mayor;

\*mayor = \*aux;

free(aux);

}