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 ejercitar el uso de **struct**.

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

En esta ocasión, no se utilizará **asignación dinámica de memoria** en ninguna función.

En esta práctica, con las estructuras dadas más abajo, se deben generar aleatoriamente los 80 alumnos de una clase**,** printando al final lo obtenido. El año de nacimiento debe estar comprendido entre 1999 y 2005. La fecha de nacimiento debe ser válida, considerando además si el año es bisiesto.

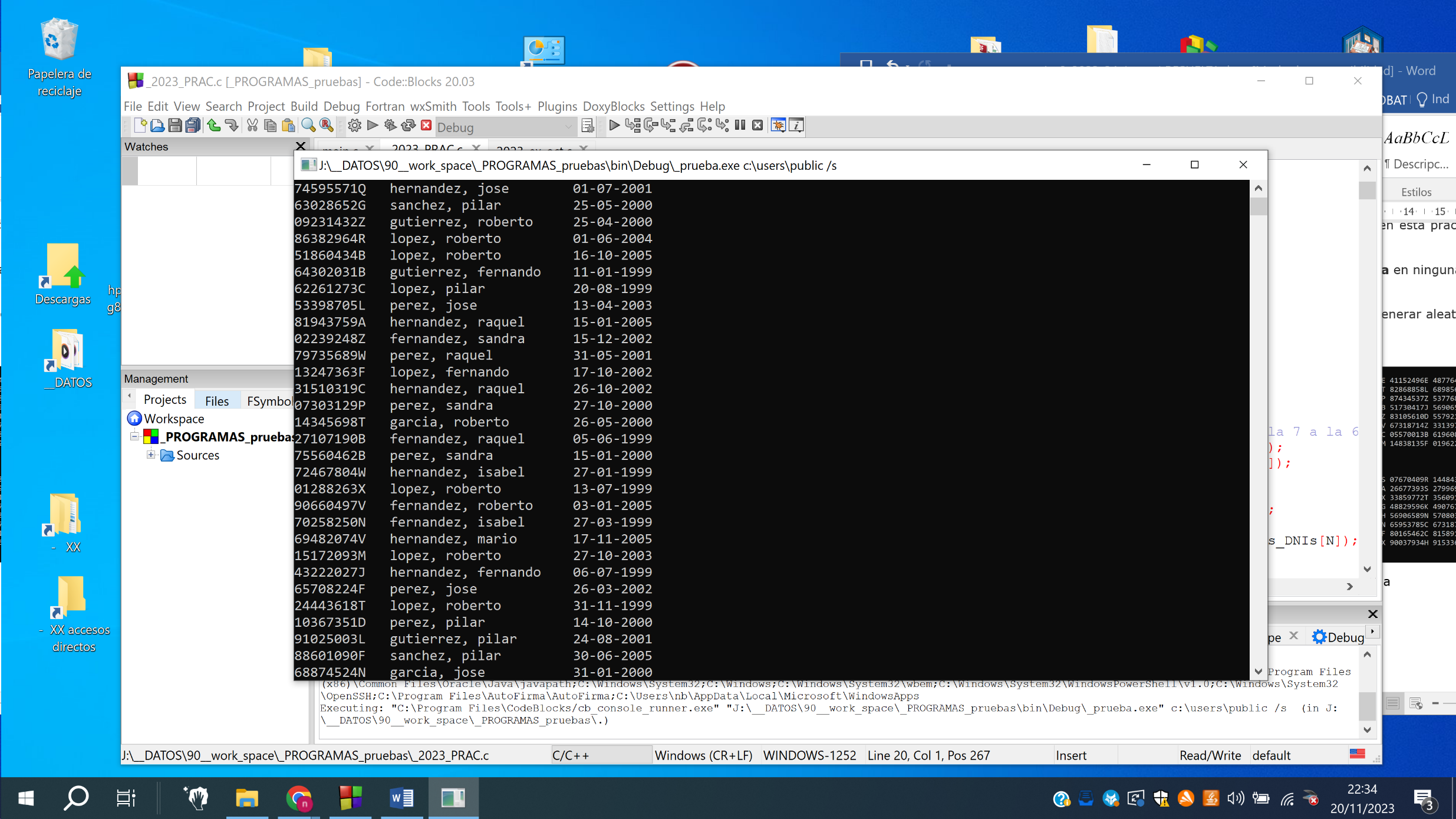


Figura 1. Ejemplo de ejecución del programa

//include

//define

#define A 80

//struct

typedef struct

{ unsigned dia, mes, anyo;

} DATE;

struct ALUMNO

{ char DNI[9+1];

char nom\_comp [20+1];

DATE nac;

};

//const

const char nombres[8][10+1]={"jose","roberto","fernando","mario",

"sandra", "raquel", "isabel", "pilar"};

const char apellidos[8][10+1]={"garcia", "hernandez", "gutierrez",

"gomez", "fernandez", "lopez", "sanchez", "perez" };

//prototipos usados en practicas anteriores

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

//prototipos de esta práctica

void print\_DATE(DATE );

void print\_ALUMNO (struct ALUMNO);

void rand\_DATE(DATE \*);

void rand\_nom\_comp(char [20+1]);

void rand\_ALUMNO(struct ALUMNO \*);

unsigned es\_fecha\_valida(DATE);

//main

int main()

{

struct ALUMNO alumnos [A];

srand(time(NULL));

int i;

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

rand\_ALUMNO(&alumnos[i]);

print\_ALUMNO(alumnos[i]);

}

return 0;

}

// definición de las funciones

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

int i,Digito,DNI = 0,resto,letraCalculada;

char letraCalculadaDNI[] = {'T', 'R', 'W', 'A', 'G', 'M', 'Y', 'F', 'P', 'D', 'X',

'B', 'N', 'J', 'Z', 'S', 'Q', 'V', 'H', 'L', 'C', 'K', 'E'};

for(i = 0;i < 9;i++){

Digito = rand() % 10;// numero del 0 - 9

DNI = DNI \* 10 + Digito;

str\_DNI[i] = Digito + '0' ;

}

resto = DNI % 23;

letraCalculada = letraCalculadaDNI[resto];

str\_DNI[8] = letraCalculada;

str\_DNI[9] = '\0';

}

unsigned es\_fecha\_valida(DATE fecha) {

unsigned Diasmes [12] = {31,28,31,30,31,30,31,31,30,31,30,31};

if ((fecha.anyo % 4 == 0 && fecha.anyo % 100 != 0) || (fecha.anyo % 400 == 0)){

Diasmes[1] = 29;

}

if(fecha.dia > Diasmes[fecha.mes - 1]){

return 0;

} else return 1;

}

void rand\_DATE(DATE \*fecha){

do{

fecha->dia = (rand()%31)+1;

fecha->mes = (rand() %12)+1;

fecha->anyo = (rand() %7)+1999;

}while(!es\_fecha\_valida(\*fecha));

}

void print\_DATE(DATE fecha){

printf("%02u - %02u - %04u\n",fecha.dia,fecha.mes,fecha.anyo);

}

void rand\_nom\_comp(char nombre\_completo[20+1]){ //strcat (concatena)

nombre\_completo [0] = '\0';

strcat(nombre\_completo,apellidos[rand()%8]);

strcat(nombre\_completo, ", ");

strcat(nombre\_completo,nombres[rand()%8]);

}

void rand\_ALUMNO(struct ALUMNO \*alumno){

rand\_str\_DNI(alumno->DNI);

rand\_nom\_comp(alumno->nom\_comp);

rand\_DATE(&alumno->nac);

}

void print\_ALUMNO(struct ALUMNO alumno) {

printf("%-15s%-32s", alumno.DNI, alumno.nom\_comp);

print\_DATE(alumno.nac);

}