|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | M.I. MARCO ANTONIO MARTINEZ QUINTANA |
| *Asignatura:* | EDA I |
| *Grupo:* | 17 |
| *No de Práctica(s):* | 4 |
| *Integrante(s):* | Colima Flores Mauricio Gibrán |
| *No. de Equipo de cómputo empleado:* |  |
| *No. de Lista o Brigada:* |  |
| *Semestre:* | 2020-2 |
| *Fecha de entrega:* | 16/02/2020 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**Objetivo de la práctica**

Utilizarás funciones en lenguaje C que permiten reservar y almacenar información de manera dinámica (en tiempo de ejecución).

**Introducción**

La memoria dinámica se refiere al espacio de almacenamiento que se reserva en tiempo de ejecución, debido a que su tamaño puede variar durante la ejecución del programa. El uso de memoria dinámica es necesario cuando a priori no se conoce el número de datos y/o elementos que se van a manejar.

**Desarrollo de la práctica**

Durante esta práctica aprenderemos a utilizar la memoria dinámica, usando las 3 funciones; malloc, calloc, free y realloc. Cada una al tener diferente funcionalidad nos ayuda saber en qué casos usarlas para poder optimizar un programa.

Programa 1: Uso de función Malloc

#include<stdio.h>

#include<stdlib.h>

int main(){

int \*arreglo, num, cont;

printf("¿Cuantos elementos tiene el conjunto?\n");

scanf("%d",&num);

arreglo=(int \*)malloc(num \* sizeof(int));

if(arreglo!=NULL){

printf("vector reservado: \n\t[");

for(cont=0; cont<num; cont++)

printf("\t%d",\*(arreglo+cont));

printf("\t]\n");

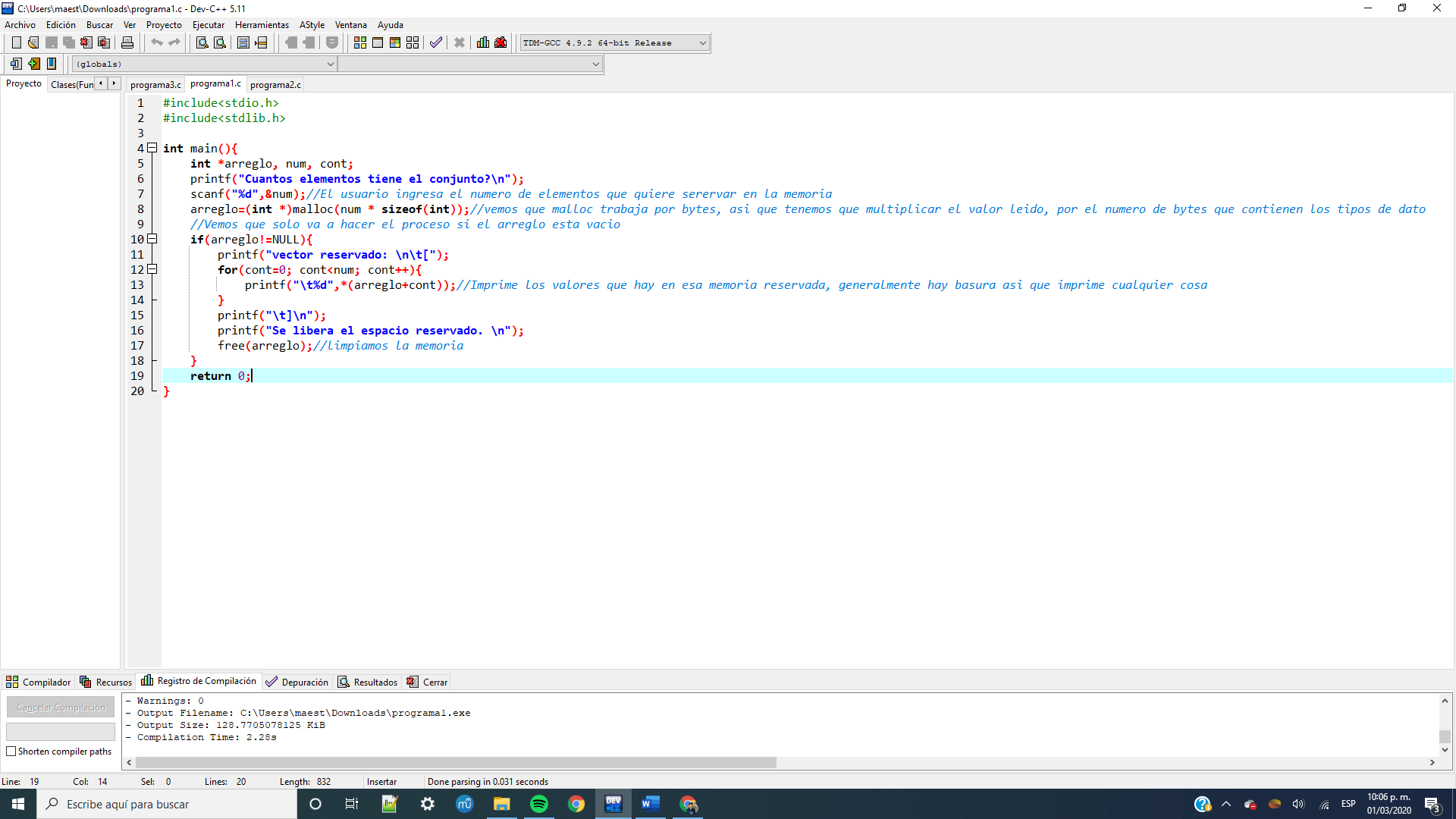
printf("Se libera el espacio reservado.\n");

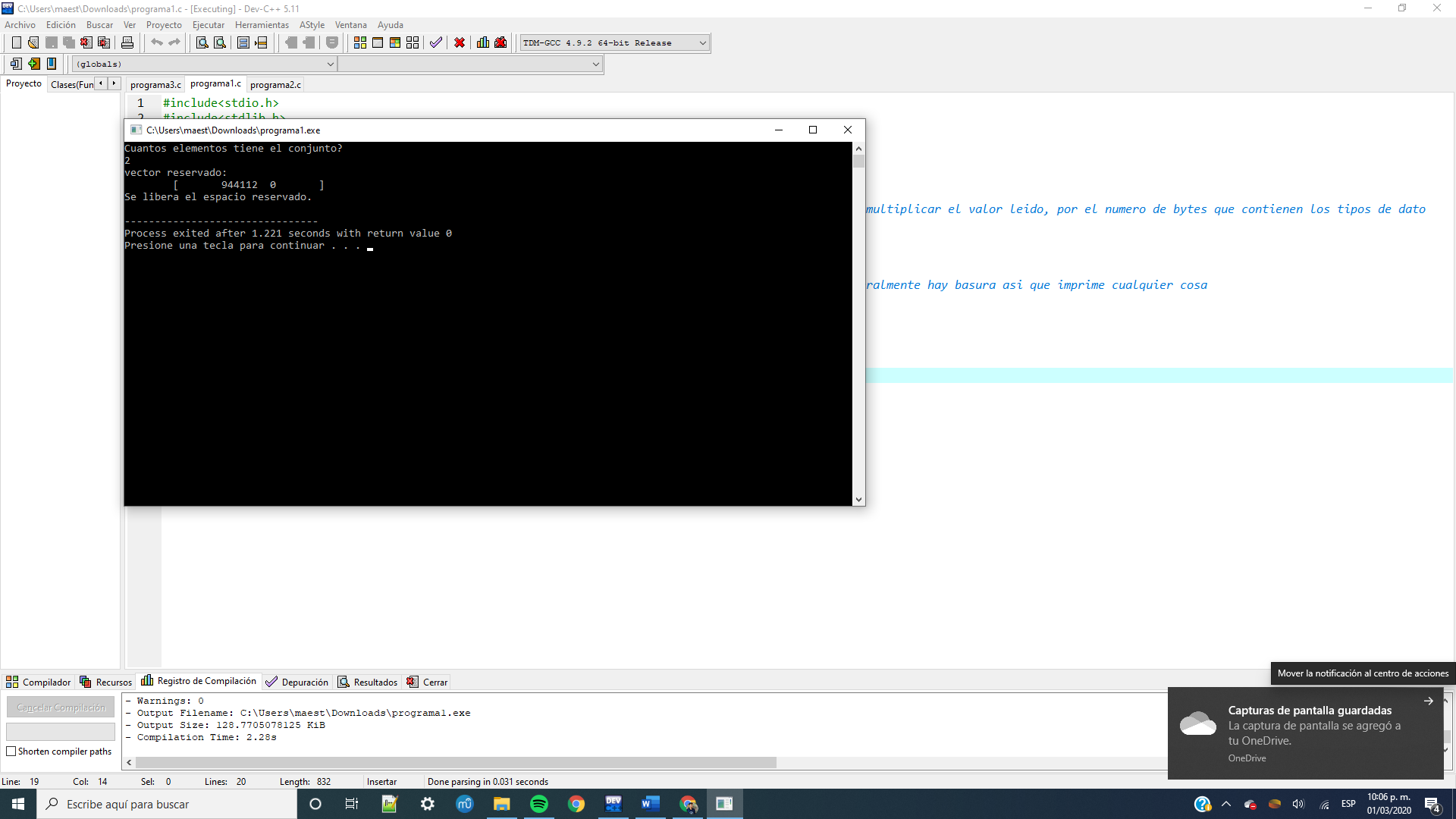
free(arreglo);

}

return 0;

}





Programa 2: Uso de función Calloc

#include<stdio.h>

#include<stdlib.h>

int main(){

int \*arreglo, num, cont;

printf("Cuantos elementos tiene el conjunto?\n");

scanf("%d",&num);

arreglo=(int \*)calloc (num, sizeof(int));

if(arreglo!=NULL){

printf("Vector reservado: \n\t[");

for(cont=0; cont<num; cont++){

printf("\t%d",\*(arreglo+cont));

}

printf("\t]\n");

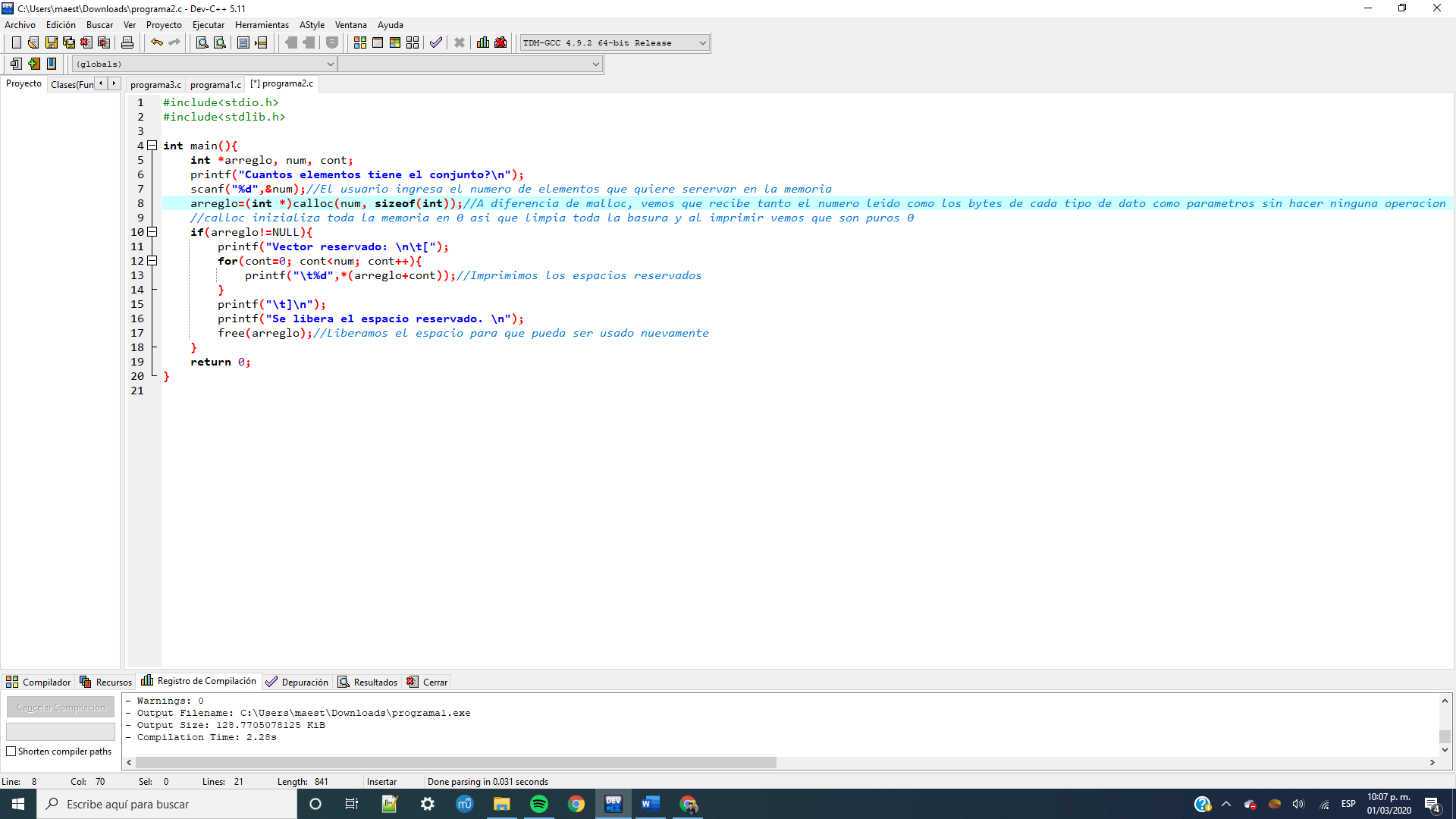
printf("Se libera el espacio reservado. \n");

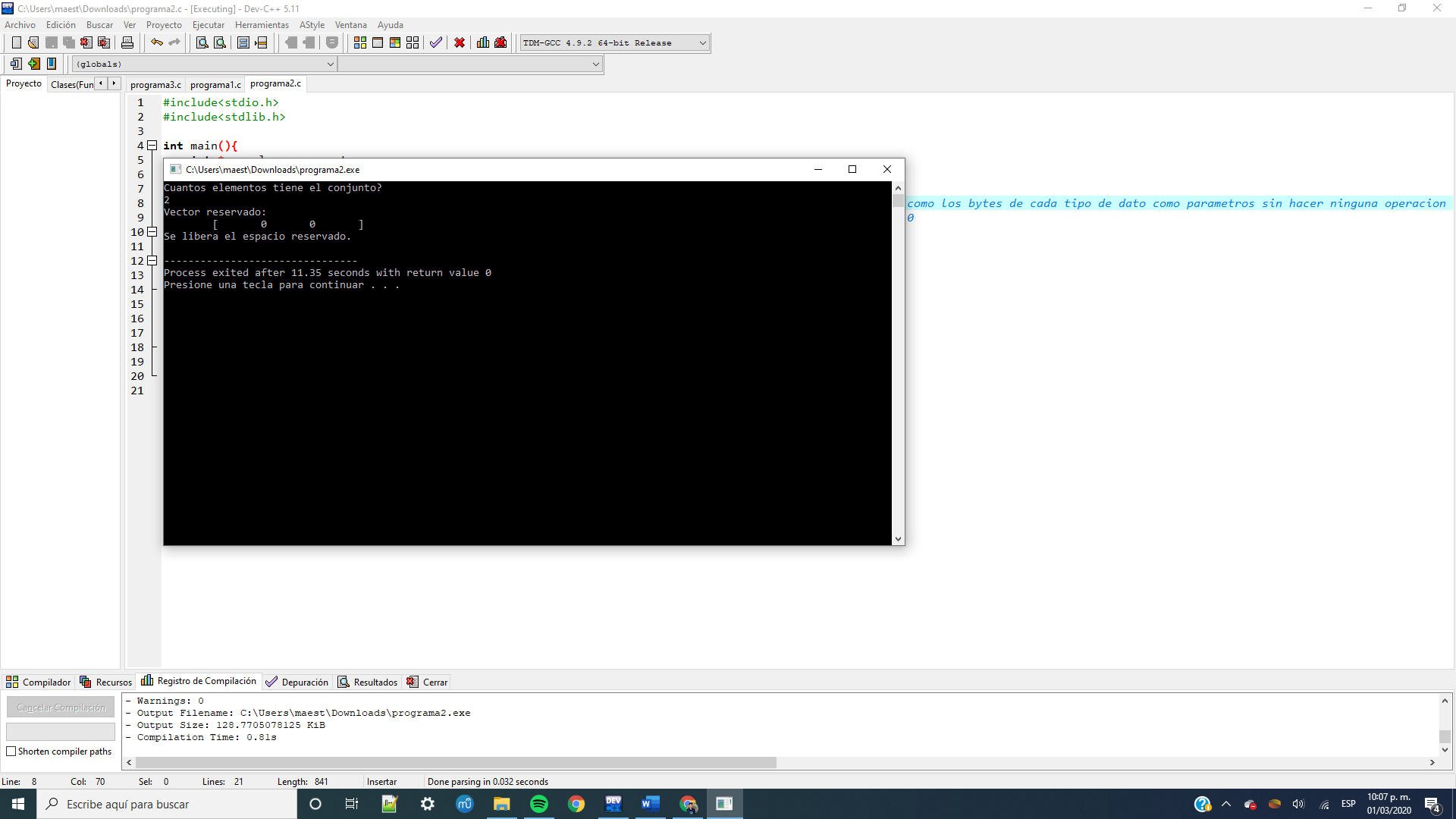
free(arreglo);

}

return 0;

}





Programa 3: Uso de función Realloc

#include<stdio.h>

#include<stdlib.h>

int main(){

int \*arreglo, \*arreglo2, num, cont;

printf("Cucntos elementos tiene el conjunto\n");

scanf("%d",&num);

arreglo=(int \*)malloc (num \* sizeof(int));

if (arreglo!=NULL){

for (cont=0;cont<num;cont++){

printf("Inserte el elemento %d del conjunto.\n",cont+1);

scanf("%d",(arreglo+cont));

}

printf("Vector insertado:\n\t[");

for(cont=0;cont<num;cont++){

printf("\t%d", \*(arreglo+cont));

}

printf("\t]\n");

printf("Aumentando el tama%co del conjunto doble.\n",164);

num \*=2;

arreglo2=(int \*)realloc(arreglo,num\*sizeof(int));

if(arreglo2!=NULL){

arreglo=arreglo2;

for(;cont<num;cont++){

printf("Inserte el elemento %d del conjunto.\n",cont+1);

scanf("%d",(arreglo2+cont));

}

printf("Vector insertado:\n\t[");

for(cont=0;cont<num;cont++){

printf("\t%d",\*(arreglo2+cont));

}

printf("\t]\n");

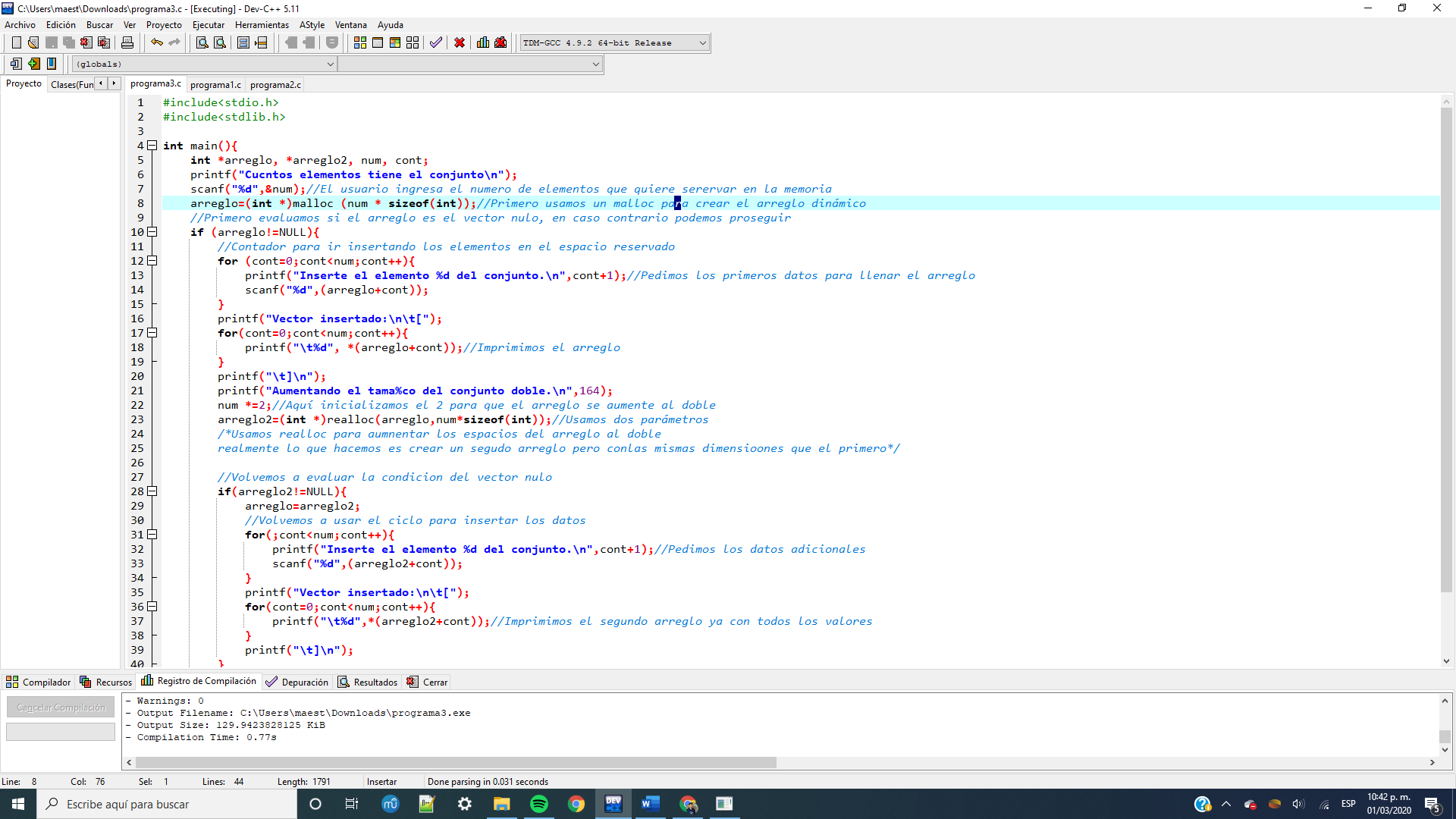
}

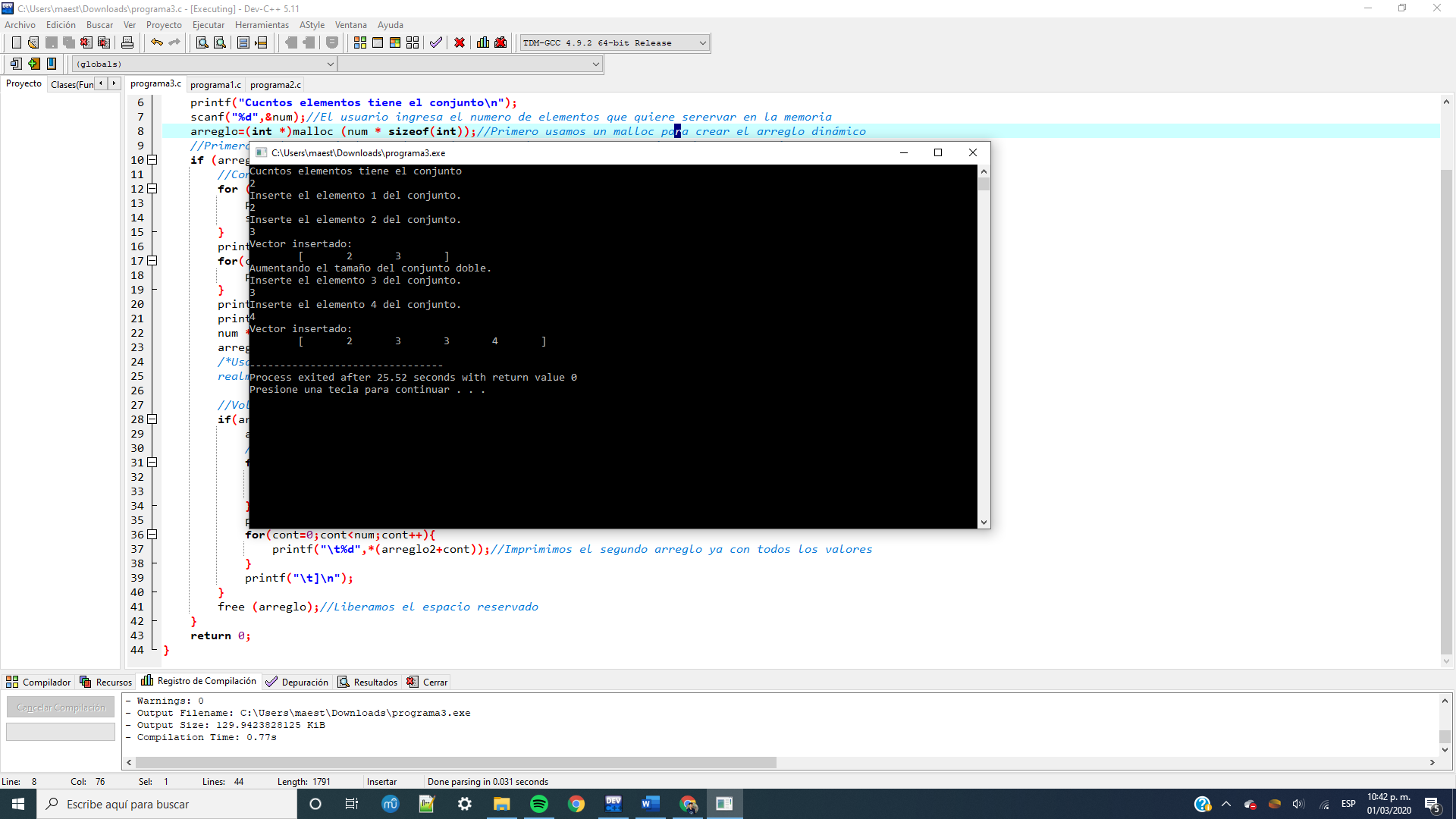
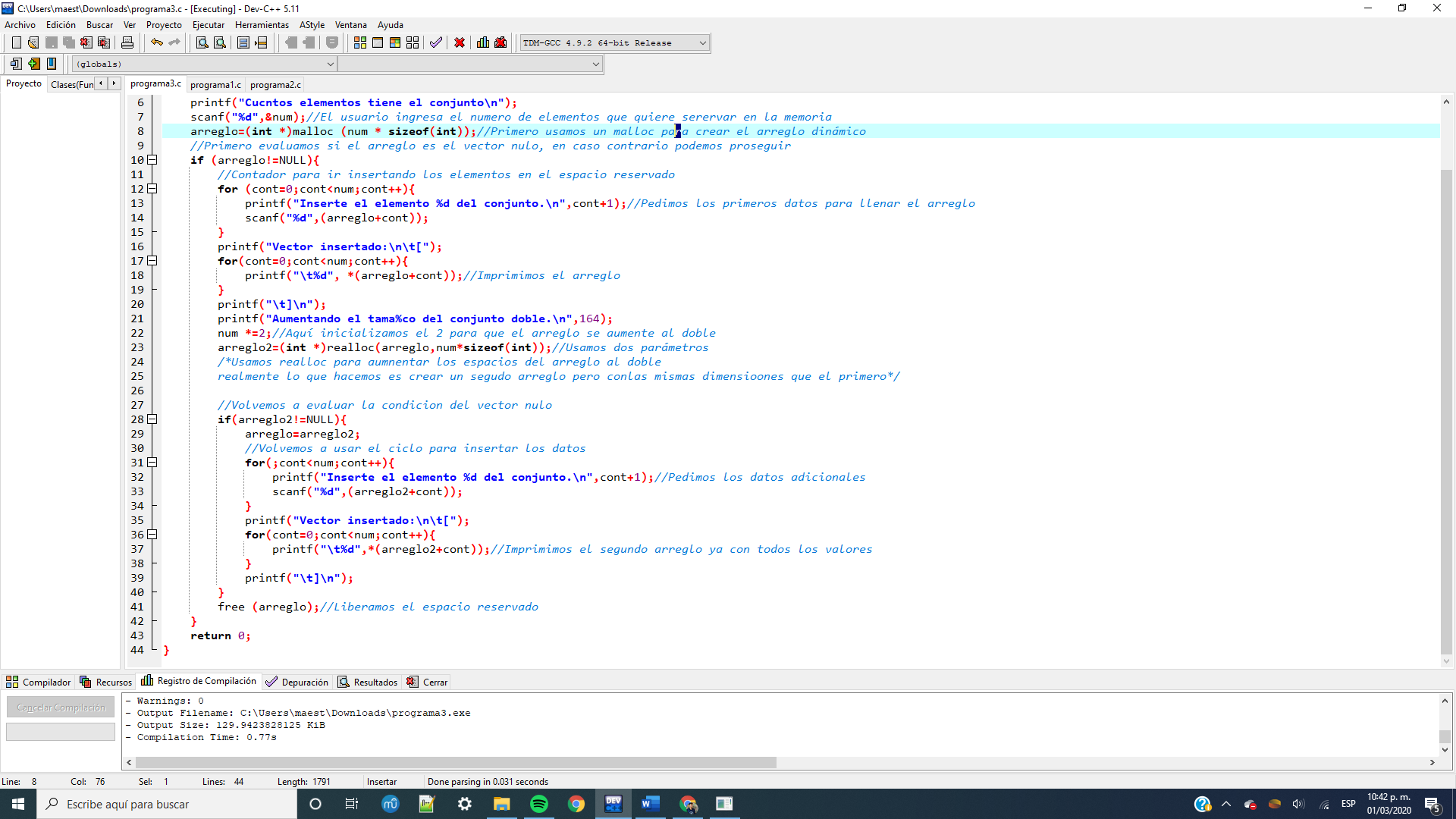
free (arreglo);

}

return 0;

}





Conclusiones:

Se lograron utilizar funciones en lenguaje C que permiten reservar y almacenar información de manera dinámica, así como aprendimos el correcto uso de las funciones Malloc, Calloc, Realloc y Free.

Bibliografía:

El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.

Ariel Rodríguez (2010). How knowing C and C++ can help you write better iPhone apps, part 1. [Figura 1]. Consulta: Enero de 2016. Disponible en: http://akosma.com/2010/10/11/how-knowing-c-and-c-can-help-you-write-betteriphone-apps-part-1/