



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: TISTA GARCÍA EDGAR

Asignatura: ESTRUCTURA DE DATOS Y ALGORITMOS I

Grupo: 1

No de Práctica(s): PRÁCTICA #4 ALMACENAMIENTO EN TIEMPO DE
EJECUCIÓN

Integrante(s): CARRILLO CERVANTES IVETTE ALEJANDRA

*No. de Equipo de
cómputo empleado:* TRABAJO EN CASA

No. de Lista o Brigada: 9

Semestre: 2021 - 2

Fecha de entrega: 18 JUNIO 2021

Observaciones:

CALIFICACIÓN: _____

PRÁCTICA #4: ALMACENAMIENTO EN TIEMPO DE EJECUCIÓN

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

Objetivo de clase:

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

Ejemplos de la guía

Código (malloc)

Este programa tiene como función principal, pedirle al usuario el número de elementos, o bien, el número de localidades de memoria que se quiere reservar utilizando la función malloc.

```
7 arreglo = (int *)malloc (num * sizeof(int));
```

malloc tiene como argumento el número de bytes a asignarse y devuelve un apuntador (tipo void) que corresponde al espacio de memoria asignada.

Como salida, muestra en pantalla las localidades de memoria que se reservaron, pero siendo la función malloc, estas localidades de memoria son espacios no consecutivos.

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P4 V1>ejemplo1.exe

¿Cuántos elementos tiene el conjunto?
3
Vector reservado:
[ 13504112      13505600      1296125778      ]
Se libera el espacio reservado.

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P4 V1>_
```

ejemplo1.c

Código (calloc)

Este programa tiene como función principal, igual que el ejemplo anterior, pedirle al usuario el número de elementos, o bien, el número de localidades de memoria que se quiere reservar utilizando ahora la función calloc.

```
7 arreglo = (int *)calloc (num, sizeof(int));
```

calloc recibe como argumento dos parámetros, el primero es el número de localidades a reservar y el segundo es el tamaño de cada una de esas localidades

Como salida, muestra en pantalla las localidades de memoria que se reservaron, pero como ahora se trata de calloc, establece ceros como valores iniciales en las localidades de memoria que reserva.

```

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P4 V1>ejemplo2.exe

¿Cuántos elementos tiene el conjunto?
3
Vector reservado:
[ 0 0 0 ]
Se libera el espacio reservado.

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P4 V1>

```

ejemplo2.c

Código(realoc)

Este programa tiene como función principal, pedirle al usuario el número de elementos, o bien, localidades de memoria que se quiere reservar mediante la función malloc, así como el valor de cada una de ellas, luego imprime los valores que el usuario inserto, para que con ayuda de la función realloc pueda aumentar el tamaño del número de localidades de memoria para reservarse al doble, y posterior a ello te pide los valores restantes e imprime todos los valores insertados anteriormente.

```

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

```

realloc recibe como argumento dos parámetros, el primero es el apuntador de tipo void previamente reservado por calloc o por malloc el cual va aumentar o disminuir y el segundo es el tamaño que queremos aumentar o disminuir

```

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P4 V1>ejemplo3.exe

¿Cuántos elementos tiene el conjunto?
3
Inserte el elemento 1 del conjunto.
6
Inserte el elemento 2 del conjunto.
16
Inserte el elemento 3 del conjunto.
4
Vector insertado:
[ 6 16 4 ]
Aumentando el tamaño del conjunto al doble.
Inserte el elemento 4 del conjunto.
7
Inserte el elemento 5 del conjunto.
12
Inserte el elemento 6 del conjunto.
32
Vector insertado:
[ 6 16 4 7 12 32 ]

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P4 V1>

```

ejemplo3.c

ACTIVIDADES PARA EL DESARROLLO DE LA PRÁCTICA

Ejercicio 1

Este programa tiene como función principal, observar las diferencias entre la función malloc y calloc. Se imprime las direcciones de memoria de un arreglo de 10 elementos de tipo entero (previamente este arreglo se inicializo con 5 elementos), así como el valor de cada elemento de este arreglo; sin embargo, como solo se inicializaron 5 elementos, los 5 restantes tienen un valor “basura”. Posterior a eso, se imprimen las direcciones y contenido del apuntador de memoria dinamica reservado por malloc (*ptr), cabe recalcar que la función malloc necesita un parámetro el cual es el número de localidades de memoria para reservar, en este caso se necesitan 10, al momento de devolver el valor del apuntador utilizado establece valores no consecutivos en las localidades de memoria que se reserva.

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P4 V1>ejercicio1.exe

DIRECCIONES DEL ARREGLO INICIALIZADO
direccion arreglo[0]=6422264    valor arreglo[0]=35
direccion arreglo[1]=6422268    valor arreglo[1]=40
direccion arreglo[2]=6422272    valor arreglo[2]=45
direccion arreglo[3]=6422276    valor arreglo[3]=50
direccion arreglo[4]=6422280    valor arreglo[4]=55
direccion arreglo[5]=6422284    valor arreglo[5]=4201104
direccion arreglo[6]=6422288    valor arreglo[6]=4200992
direccion arreglo[7]=6422292    valor arreglo[7]=7612960
direccion arreglo[8]=6422296    valor arreglo[8]=7612912
direccion arreglo[9]=6422300    valor arreglo[9]=9

DIRECCIONES Y CONTENIDO DEL APUNTADOR DE MEMORIA DINAMICA RESERVADO CON MALLOC
direccion=7612912    *valor=7610592
direccion=7612916    *valor=7612856
direccion=7612920    *valor=0
direccion=7612924    *valor=0
direccion=7612928    *valor=0
direccion=7612932    *valor=0
direccion=7612936    *valor=0
direccion=7612940    *valor=0
direccion=7612944    *valor=0
direccion=7612948    *valor=0

Presione una tecla para continuar . . .

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P4 V1>
```

Captura del programa antes de modificarlo

Modificamos la función malloc por la función calloc, recordando que para utilizar esta función se necesitan dos parámetros, el primero es el número de localidades a reservar y el segundo es el tamaño de cada una de ellas, por lo tanto la función queda de la siguiente manera.

```
15      int *ptr = calloc(10, sizeof(int));
```

Función modificada de malloc a calloc.

y nos muestra la salida siguiente:

```

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P4 V1>ejercicio1.exe

DIRECCIONES DEL ARREGLO INICIALIZADO
direccion arreglo[0]=6422264    valor arreglo[0]=35
direccion arreglo[1]=6422268    valor arreglo[1]=40
direccion arreglo[2]=6422272    valor arreglo[2]=45
direccion arreglo[3]=6422276    valor arreglo[3]=50
direccion arreglo[4]=6422280    valor arreglo[4]=55
direccion arreglo[5]=6422284    valor arreglo[5]=4201120
direccion arreglo[6]=6422288    valor arreglo[6]=4201008
direccion arreglo[7]=6422292    valor arreglo[7]=6564384
direccion arreglo[8]=6422296    valor arreglo[8]=6564336
direccion arreglo[9]=6422300    valor arreglo[9]=9

DIRECCIONES Y CONTENIDO DEL APUNTADOR DE MEMORIA DINAMICA RESERVADO CON CALLOC
direccion=6564336    *valor=0
direccion=6564340    *valor=0
direccion=6564344    *valor=0
direccion=6564348    *valor=0
direccion=6564352    *valor=0
direccion=6564356    *valor=0
direccion=6564360    *valor=0
direccion=6564364    *valor=0
direccion=6564368    *valor=0
direccion=6564372    *valor=0

Presione una tecla para continuar . . .

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P4 V1>

```

ejercicio1.c

En este caso al ser la función calloc, cuando devuelve el apuntador utilizado, establece ceros como valores iniciales en las localidades de memoria que reserva.

Se observa que la principal diferencia entre la función malloc y la función calloc, es al momento de devolver el apuntador que se utiliza, ya que calloc establece como ceros los valores iniciales de la memoria que reserva. Se supone que las direcciones de memoria deberían de haber sido las mismas en ambos casos; sin embargo, el compilador que utilices puede influir en esto.

Ahora, agregamos una instrucción para asignar valores en las localidades de memoria reservadas con memoria dinamica, estos valores serán los múltiplos de 4, para realizar esta instrucción, se necesito uso de un ciclo for para poder modificar cada uno de los valores del arreglo.

```

49      *(ptr + cont) = 4*(cont+1);
50      // *(PTR + CONT) LUGARES DE MEMORIA PARA ASIGNAR EJ. *(PTR + 0) <- SE GUARDA EN LA POSICION 0 DE LA MEMORIA
51      // 4*(CONT + 1) QUEREMOS QUE EMPIECE DESDE EL 4, NO DEL CERO POR LO QUE A CONT SUMAMOS 1 Y LO MULTIPLICAMOS POR 4

```

Se asignan valores en las localidades de memoria reservadas con múltiplos de 4

La salida queda como:

```

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P4 V1>ejercicio1.exe

DIRECCIONES DEL ARREGLO INICIALIZADO
direccion arreglo[0]=6422264    valor arreglo[0]=35
direccion arreglo[1]=6422268    valor arreglo[1]=40
direccion arreglo[2]=6422272    valor arreglo[2]=45
direccion arreglo[3]=6422276    valor arreglo[3]=50
direccion arreglo[4]=6422280    valor arreglo[4]=55
direccion arreglo[5]=6422284    valor arreglo[5]=4201280
direccion arreglo[6]=6422288    valor arreglo[6]=4201168
direccion arreglo[7]=6422292    valor arreglo[7]=7088672
direccion arreglo[8]=6422296    valor arreglo[8]=7088624
direccion arreglo[9]=6422300    valor arreglo[9]=9

DIRECCIONES Y CONTENIDO DEL APUNTADOR DE MEMORIA DINAMICA RESERVADO CON CALLOC
direccion=7088624    *valor=0
direccion=7088628    *valor=0
direccion=7088632    *valor=0
direccion=7088636    *valor=0
direccion=7088640    *valor=0
direccion=7088644    *valor=0
direccion=7088648    *valor=0
direccion=7088652    *valor=0
direccion=7088656    *valor=0
direccion=7088660    *valor=0

DIRECCION DE MEMORIA Y EL VALOR DE CADA ELEMENTO (MULTIPLS DE 4)
direccion=7088624    *valor=4
direccion=7088628    *valor=8
direccion=7088632    *valor=12
direccion=7088636    *valor=16
direccion=7088640    *valor=20
direccion=7088644    *valor=24
direccion=7088648    *valor=28
direccion=7088652    *valor=32
direccion=7088656    *valor=36
direccion=7088660    *valor=40

Presione una tecla para continuar . . .

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P4 V1>

```

Se observa que las direcciones de memoria no cambian, sigue siendo las mismas, aun cuando asignamos valores a las localidades de memoria reservadas por calloc.

Para concluir este ejercicio, se agrega la función realloc, para aumentar las localidades de memoria reservada, utilizando dos tipos de apuntadores, el primero es el mismo que se utilizó para reservar la memoria y el otro es un apuntador nuevo.

```

71      ptr = (int*)realloc(ptr, sizeof(int)*20);

```

Utilizamos la función realloc con el mismo apuntador

```

82      int *ptr3 = (int*)realloc(ptr, 20*sizeof(int)); // ES LA MISMA LOCALIDAD DE MEMORIA

```

Utilizamos la función realloc con un nuevo apuntador

Esto muestra en pantalla lo siguiente:


```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P4 V1>ejercicio1.exe
```

DIRECCIONES DEL ARREGLO INICIALIZADO

```
direccion arreglo[0]=6422240    valor arreglo[0]=35
direccion arreglo[1]=6422244    valor arreglo[1]=40
direccion arreglo[2]=6422248    valor arreglo[2]=45
direccion arreglo[3]=6422252    valor arreglo[3]=50
direccion arreglo[4]=6422256    valor arreglo[4]=55
direccion arreglo[5]=6422260    valor arreglo[5]=2091685970
direccion arreglo[6]=6422264    valor arreglo[6]=-2
direccion arreglo[7]=6422268    valor arreglo[7]=6422280
direccion arreglo[8]=6422272    valor arreglo[8]=1992847149
direccion arreglo[9]=6422276    valor arreglo[9]=4201456
```

DIRECCIONES Y CONTENIDO DEL APUNTADEOR DE MEMORIA DINAMICA RESERVADO CON CALLOC

```
direccion=7154160    *valor=0
direccion=7154164    *valor=0
direccion=7154168    *valor=0
direccion=7154172    *valor=0
direccion=7154176    *valor=0
direccion=7154180    *valor=0
direccion=7154184    *valor=0
direccion=7154188    *valor=0
direccion=7154192    *valor=0
direccion=7154196    *valor=0
```

DIRECCION DE MEMORIA Y EL VALOR DE CADA ELEMENTO (MULTIPLICOS DE 4)

```
direccion=7154160    *valor=4
direccion=7154164    *valor=8
direccion=7154168    *valor=12
direccion=7154172    *valor=16
direccion=7154176    *valor=20
direccion=7154180    *valor=24
direccion=7154184    *valor=28
direccion=7154188    *valor=32
direccion=7154192    *valor=36
direccion=7154196    *valor=40
```

REALLOC

UTILIZAMOS EL MISMO APUNTADEOR PTR

```
Direccion = 7151840    *valor = 4
Direccion = 7151844    *valor = 8
Direccion = 7151848    *valor = 12
Direccion = 7151852    *valor = 16
Direccion = 7151856    *valor = 20
Direccion = 7151860    *valor = 24
Direccion = 7151864    *valor = 28
Direccion = 7151868    *valor = 32
Direccion = 7151872    *valor = 36
Direccion = 7151876    *valor = 40
Direccion = 7151880    *valor = 0
Direccion = 7151884    *valor = 0
Direccion = 7151888    *valor = 0
Direccion = 7151892    *valor = 0
Direccion = 7151896    *valor = 1992511808
Direccion = 7151900    *valor = 0
Direccion = 7151904    *valor = 0
Direccion = 7151908    *valor = 0
Direccion = 7151912    *valor = 1992511808
Direccion = 7151916    *valor = 0
```

UTILIZAMOS UN APUNTADEOR DIFERENTE PTR3

```
Direccion = 7151840    *valor = 4
Direccion = 7151844    *valor = 8
Direccion = 7151848    *valor = 12
Direccion = 7151852    *valor = 16
Direccion = 7151856    *valor = 20
Direccion = 7151860    *valor = 24
Direccion = 7151864    *valor = 28
Direccion = 7151868    *valor = 32
Direccion = 7151872    *valor = 36
Direccion = 7151876    *valor = 40
Direccion = 7151880    *valor = 0
Direccion = 7151884    *valor = 0
Direccion = 7151888    *valor = 0
Direccion = 7151892    *valor = 0
Direccion = 7151896    *valor = 1992511808
Direccion = 7151900    *valor = 0
Direccion = 7151904    *valor = 0
Direccion = 7151908    *valor = 0
Direccion = 7151912    *valor = 1992511808
Direccion = 7151916    *valor = 0
```

Presione una tecla para continuar . . .

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P4 V1>
```

ejercicio1.c

Observamos que al utilizar la función realloc con el mismo apuntador (ptr) y al utilizar un nuevo apuntador (ptr3), las direcciones de memoria, así como sus valores son exactamente igual.

Ejercicio 2

Se observa que el archivo "Alumno.h" es un tipo de dato abstracto, ya que cuenta con dos estructuras anidadas, una estructura de tipo Dirección (sus miembros son los datos del domicilio del alumno) y la otra estructura es de tipo Alumno (los miembros son los datos del alumno). Al ejecutar el archivo "ejercicio2.c" tiene como salida el tamaño de la estructura Alumno, el cual es el valor de 88, esto se debe a que suma el tamaño de cada miembro de la estructura (se sabe que el tamaño de byte de un entero es 4, de un flotante es 4 y de un carácter es 1); después imprime la dirección de memoria del primer y segundo apuntador respectivamente, para que al final con la función realloc incrementa el tamaño de localidades de memoria del segundo apuntador

Después se agregaron instrucciones para pedir al usuario los datos de los alumnos que se reservan en el programa, esto se logró con ayuda de funciones (crear alumno, llenarAlumno, imprimirAlumno).

Finalmente observamos que podemos liberar la memoria reservada por la función calloc o malloc con la función free, pero también con la función realloc, ya que esta última sabemos que sirve tanto para aumentar como para disminuir espacios de memoria, por lo que indicamos que queremos que disminuya el valor de espacios reservados a cero y para comprobar que si se libera la memoria, después de realizar esta función volvemos a imprimir los datos obtenidos del alumno y no nos muestra nada en pantalla.

```
73     din1 = (Alumno *)realloc(din1, 0);
74
75     din2 = (Alumno *)realloc(din1, 0);
```

Cambiamos free por realloc

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P4 V2>ejercicio2.exe
```

```
Tamaño de objeto Alumno = 88
```

```
Primer apuntador:
```

```
Direccion[0] = 2048624
Direccion[1] = 2048712
Direccion[2] = 2048800
Direccion[3] = 2048888
Direccion[4] = 2048976
```

```
Segundo apuntador
```

```
Direccion[0] = 2049104
Direccion[1] = 2049192
Direccion[2] = 2049280
Direccion[3] = 2049368
Direccion[4] = 2049456
```

```
Con realloc:
```

```
&din3[0] = 2049104
&din3[1] = 2049192
&din3[2] = 2049280
&din3[3] = 2049368
&din3[4] = 2049456
&din3[5] = 2049544
&din3[6] = 2049632
&din3[7] = 2049720
&din3[8] = 2049808
&din3[9] = 2049896
```

```
DATOS DEL ALUMNO
```

```
Num de Cuenta: 3123456
```

```
Nombre: Forysthe
```

```
Apellido: Pendleton Jones
```

```
Promedio: 9
```

```
DIRECCION
```

```
Calle: Elm Street
```

```
Numero: 111
```

```
Colonia: Riverdale
```


Codigo Postal: 20737

INFORMACION PERSONAL
Numero de cuenta: 3123456
Name: Forysthe
LastName: Pendleton Jones
Promedio: 9.00

DIRECCION
Calle: Elm Street
Numero: 111
Colonia: Riverdale
Codigo Postal: 20737

DATOS DEL ALUMNO

Num de Cuenta: 3216549
Nombre: Veronica
Apellido: Lodge
Promedio: 9.6

DIRECCION

Calle: Pembroke
Numero: 305
Colonia: River Dale
Codigo Postal: 20738

INFORMACION PERSONAL
Numero de cuenta: 3216549
Name: Veronica
LastName: Lodge
Promedio: 9.60

DIRECCION
Calle: Pembroke
Numero: 305
Colonia: River Dale
Codigo Postal: 20738

DATOS DEL ALUMNO CON DIN1
Numero de cuenta: 3123456
Nombre: Forysthe
Apellido: Pendleton Jones
Promedio: 9.00

DATOS DEL ALUMNO CON DIN2
Numero de cuenta: 3216549
Nombre: Veronica
Apellido: Lodge
Promedio: 9.60

MEMORIA LIBERADA

DATOS DEL ALUMNO CON DIN1

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P4 V2>gcc ejercicio2.c -o ejercicio2.exe

ejercicio2.c

Ejercicio 3

Para este último ejercicio se diseñó un tipo de dato abstracto llamado computadora el cual tiene como miembros los datos de una computadora (como son la marca, modelo, procesador, etc.), esta estructura se guardó en un archivo head llamado "Computadora.h" y se mandó a llamar desde el archivo "ejercicio3.c"; se realizó un "arreglo dinámico de computadoras" con la función `calloc`, se decidió usar esta función ya que recibe dos parámetros, el primero es el número de localidades a reservar (la cantidad que ingresó el usuario) y el segundo es el tamaño de cada una de estas localidades.

```
18 Computadora *creadas = (Computadora *)calloc(cantidad, sizeof(Computadora));
```

Función calloc ocupada

el cual solicita al usuario el tamaño del arreglo para después llenar los datos de cada computadora ingresada, posterior a ello se imprimen los datos de cada computadora. Esto se logró con ayuda de diferentes funciones (`llenarDatosComputadora`, `imprimirDatosComputadora`)

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P4 V2>ejercicio3.exe

¿CUANTAS COMPUTADORAS QUIERES CREAR? 2

INGRESE LOS DATOS DE LAS COMPUTADORAS

COMPUTADORA 1
Marca: HP
Modelo: 6QW59LA
Procesador: RYZEN
Tipo de almacenamiento: SSD
Capacidad de memoria RAM: 16
Capacidad de memoria GB: 476

COMPUTADORA 2
Marca: APPLE
Modelo: MVVJ2E/A
Procesador: INTEL
Tipo de almacenamiento: SSD
Capacidad de memoria RAM: 16
Capacidad de memoria GB: 512
DATOS DE LAS COMPUTADORAS

COMPUTADORA 1
Marca: HP
Modelo: 6QW59LA
Procesador: RYZEN
Tipo de almacenamiento: SSD
RAM: 16 GB
GB: 476 GB

COMPUTADORA 2
Marca: APPLE
Modelo: MVVJ2E/A
Procesador: INTEL
Tipo de almacenamiento: SSD
RAM: 16 GB
GB: 512 GB

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Carrillo Cervantes
Ivette Alejandra G1 P4 V2>
```

ejercicio3.c

Conclusiones

Se cumplieron los objetivos de esta práctica, ya que se hizo uso de memoria dinamica la cual permitió reservar y almacenar información en cada uno de estos ejercicios, aparte de que retomamos temas antes vistos en clase como son los apuntadores los cuales en memoria dinamica tienen la función de guardar información en los espacios reservados, y también se retomó el tema de funciones que fue lo que más se utilizó para trabajar con implementaciones de tipo abstracto de datos.

En esta práctica se logró cubrir el 100 % de los ejercicios solicitados, aunque al principio me costó un poco de trabajo, al volver a ver los videos de las clases de memoria dinamica e investigando, me quedo más clara la idea y los pude realizar.

Considero que estos ejercicios si contribuyeron al aprendizaje del concepto visto en clase, ya que se utilizaron las funciones de memoria dinamica (malloc, calloc, realloc y free) y se vio sus diferentes usos, por ejemplo, algo que no tenía en cuenta hasta que lo pidió la práctica era que la función realloc también se utiliza para liberar memoria. Estos ejercicios estuvieron muy bien planteados, ya que son un buen ejemplo de como se utiliza la memoria dinamica. Personalmente, siento que la principal ventaja de utilizar memoria dinamica a la hora de programar, es que puedes aumentar o disminuir el tamaño de localidades de memoria reservada durante la ejecución del programa.

Referencias

- Joyanes. Programación en C. Departamento de Lenguajes y Sistemas Informáticos e Ingeniería de Software. 2da Edición