# Objetivo de la práctica

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

# Ejercicios de la guía de laboratorio

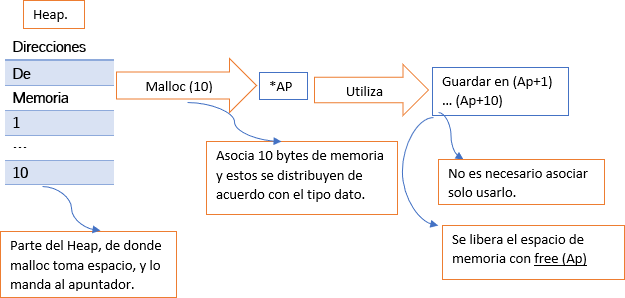
* Primer ejemplo: Código (malloc).

En este código se utiliza la función malloc que permite el uso de memoria dinámica en los programas de c, esta se caracteriza por utilizar un apuntador para apuntar a un conjunto de memoria disponible del heap. Es necesario para esta función el indicar la cantidad de bytes del tipo de dato e indicar cuantos bytes de ese dato son necesarios.

Además, se debe indicar la conversión del tipo de dato ya que malloc regresa un tipo de dato void, que no está asociado a algún tipo de dato.



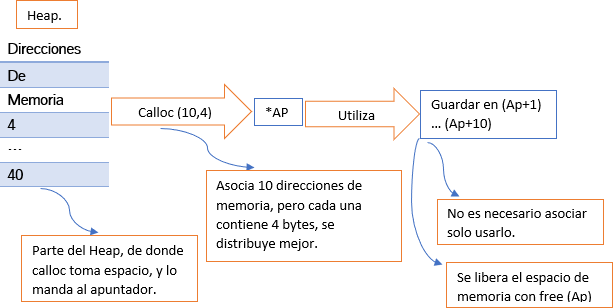
Diagrama de funcionamiento.



* Segundo ejemplo: Código (calloc).

En este código se emplear conceptualmente el mismo hecho de utilizar una función llamada calloc para poder asignar memoria dinámica a un apuntador y después utilizarlo, en este caso los argumentos de la función son dos, uno corresponde al número de bytes y el segundo corresponde a la cantidad de bytes por cada uno de los bytes que se colocaron primero o en otras palabras corresponden a la cantidad de bytes de acuerdo con el tipo de dato.

Diagrama de funcionamiento.

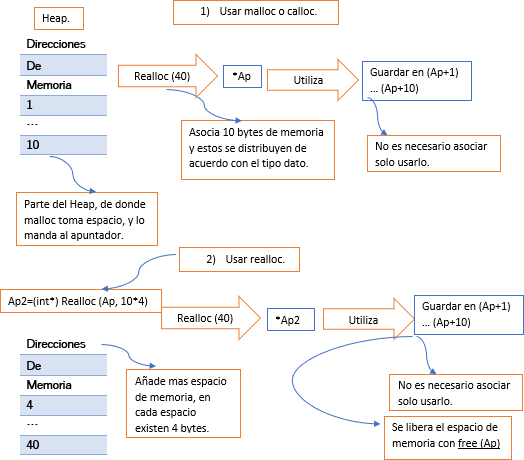


* Tercer ejemplo: Código (realloc).

En este código se utiliza una tercera función de memoria dinámica que permite aumentar el tamaño de bytes que se reservan con las funciones malloc y calloc, lo que ayuda a tener a disposición más memoria para trabajar, la forma en que funciona realloc es por medio de dos apuntadores, el primero corresponde al apuntador que asocio malloc o calloc a los bytes de memoria, el segundo es un apuntador que almacenara al primero,

además el espacio extra asignado. Se sigue una consideración para utilizar realloc es que necesita que se indique todos los bytes que se van a necesitar más los que ya se tenían apartados, de lo contario podría reducir la cantidad de bytes de memoria.

Diagrama de funcionamiento.



Preguntas

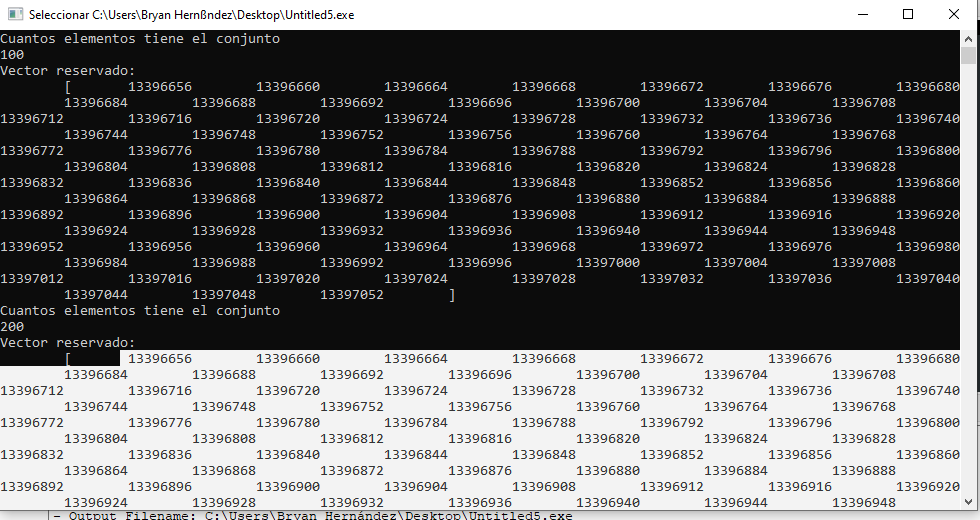
* ¿Si se usa realloc en el mismo apuntador funciona?

Funciona cuando no contienen nada las direcciones de memoria, pero incluso si se inserta algo antes es necesario hacer una igualación para conservar los datos, no es necesariamente obligatorio porque conserva los datos.



* ¿Cuándo se asigna más memoria dinámica con realloc este busca dentro de las mismas localidades de memoria o se mueve a nuevas localidades?

Al hacer pruebas si va sobre las mismas localidades de memoria, pero siento que acierta cantidad de bytes se va a saltar a otras localidades.



* Con la instrucción (int \*) malloc (sizeof(int)\*3), sin el operador (\*), ¿funciona?

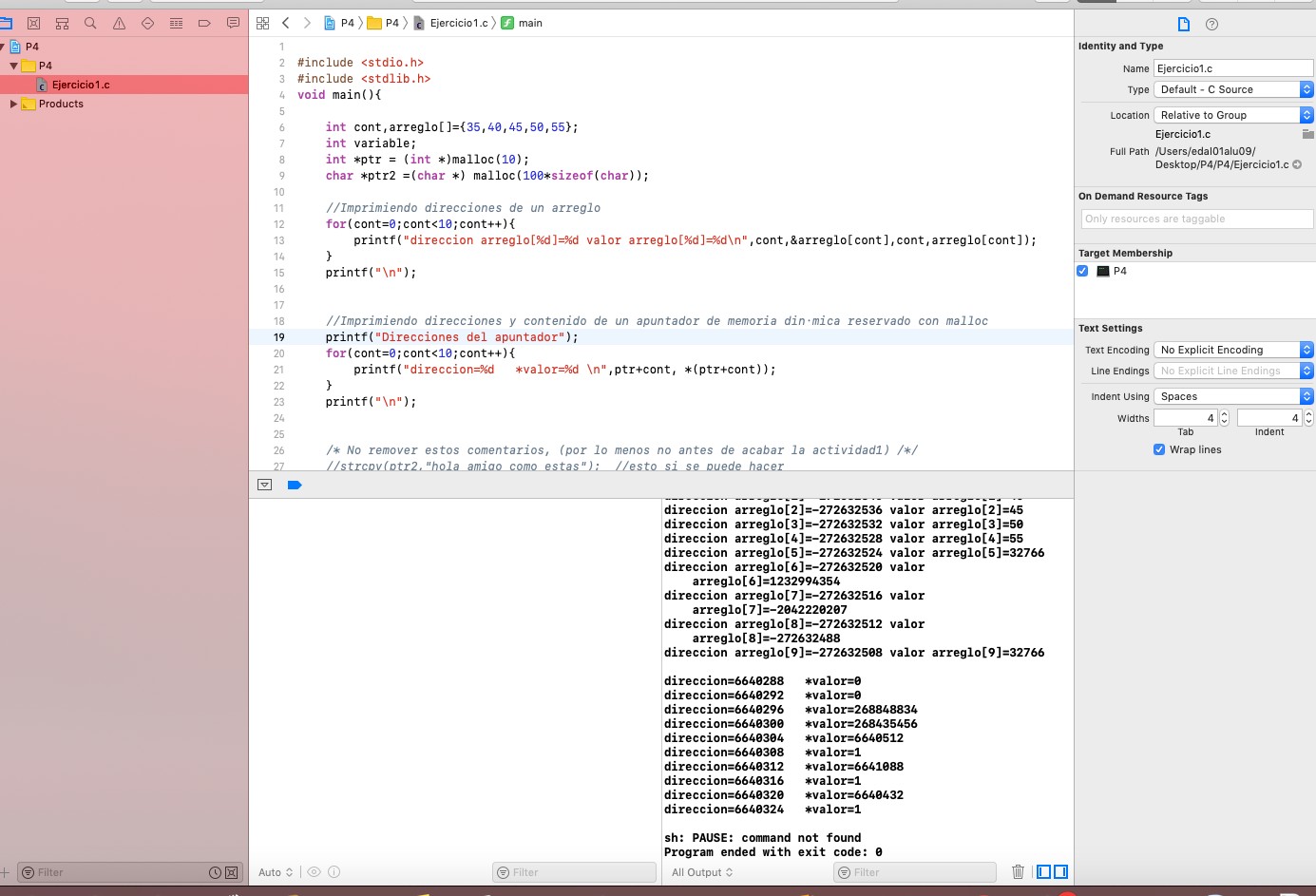
No funciona porque detecta únicamente el valor como constante y es necesario indicar que hace la constante.



# Ejercicios del Laboratorio

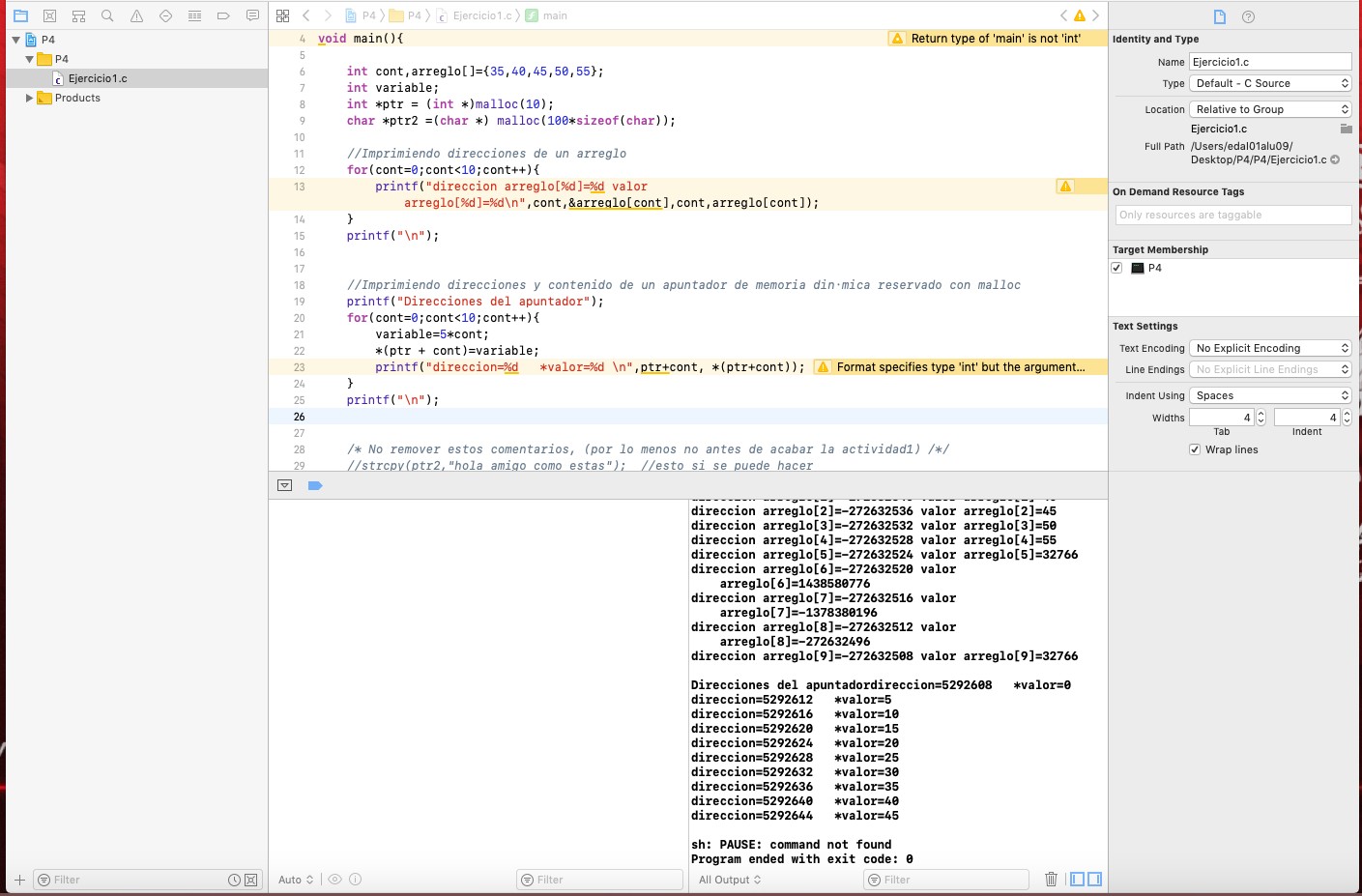
Primer ejercicio.

1. En las dalias del programa se observan las direcciones de memoria que se le han asignado al apuntador, en por medio de la función malloc la cual no limpia le contenido de esta direcciones por ello aparecen valores desconocidos.
2. Además se ven los valores de memoria que tiene un arreglo y el contenido del mismo, sin embargo ademas imprime direcciones y valores que no están declarados dentro del arreglo



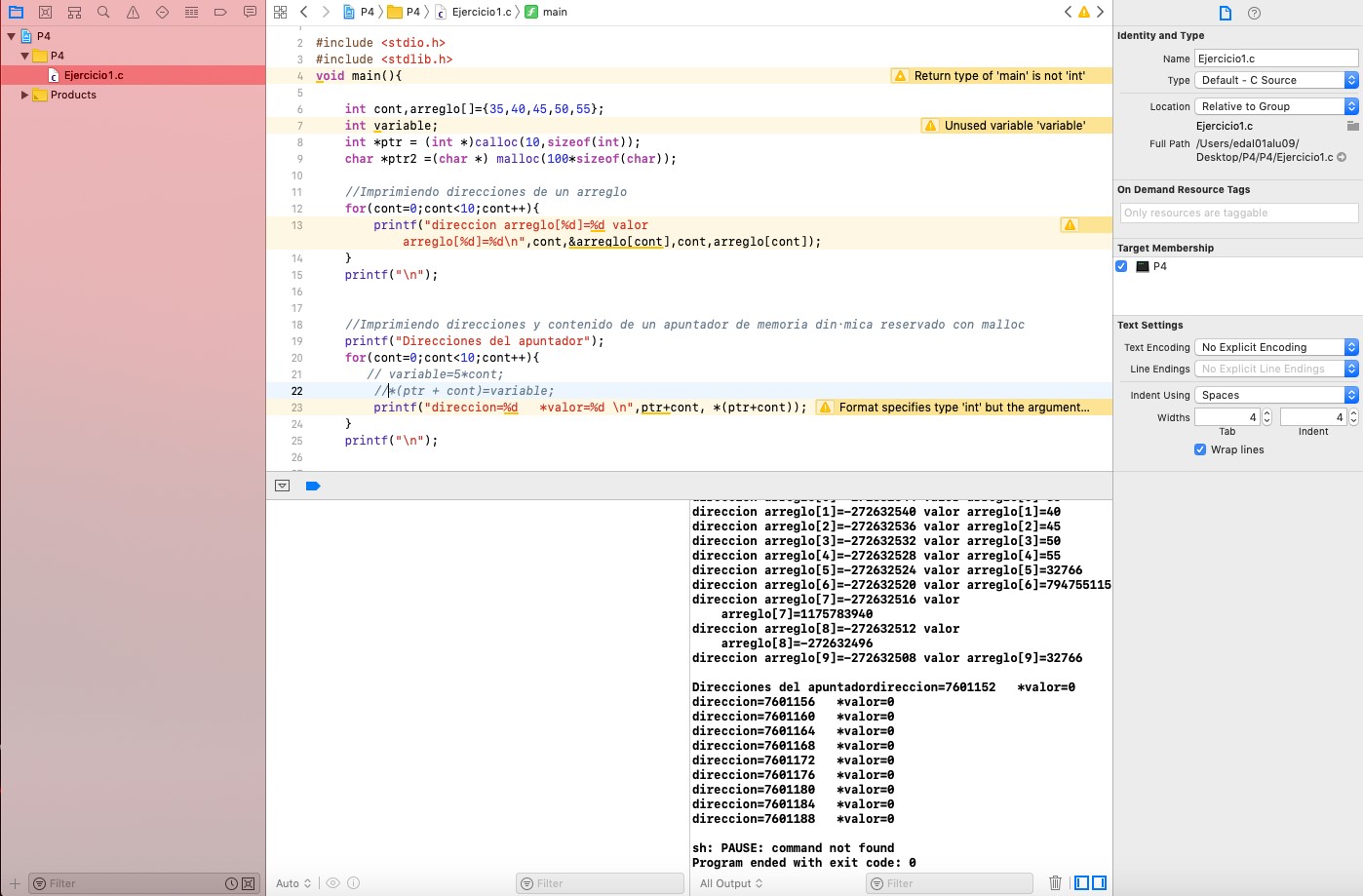
Segundo inciso.

Se imprimen los valores de la dirección de memoria donde se guardan los valores de los múltiplos del 5.



Tercer inciso

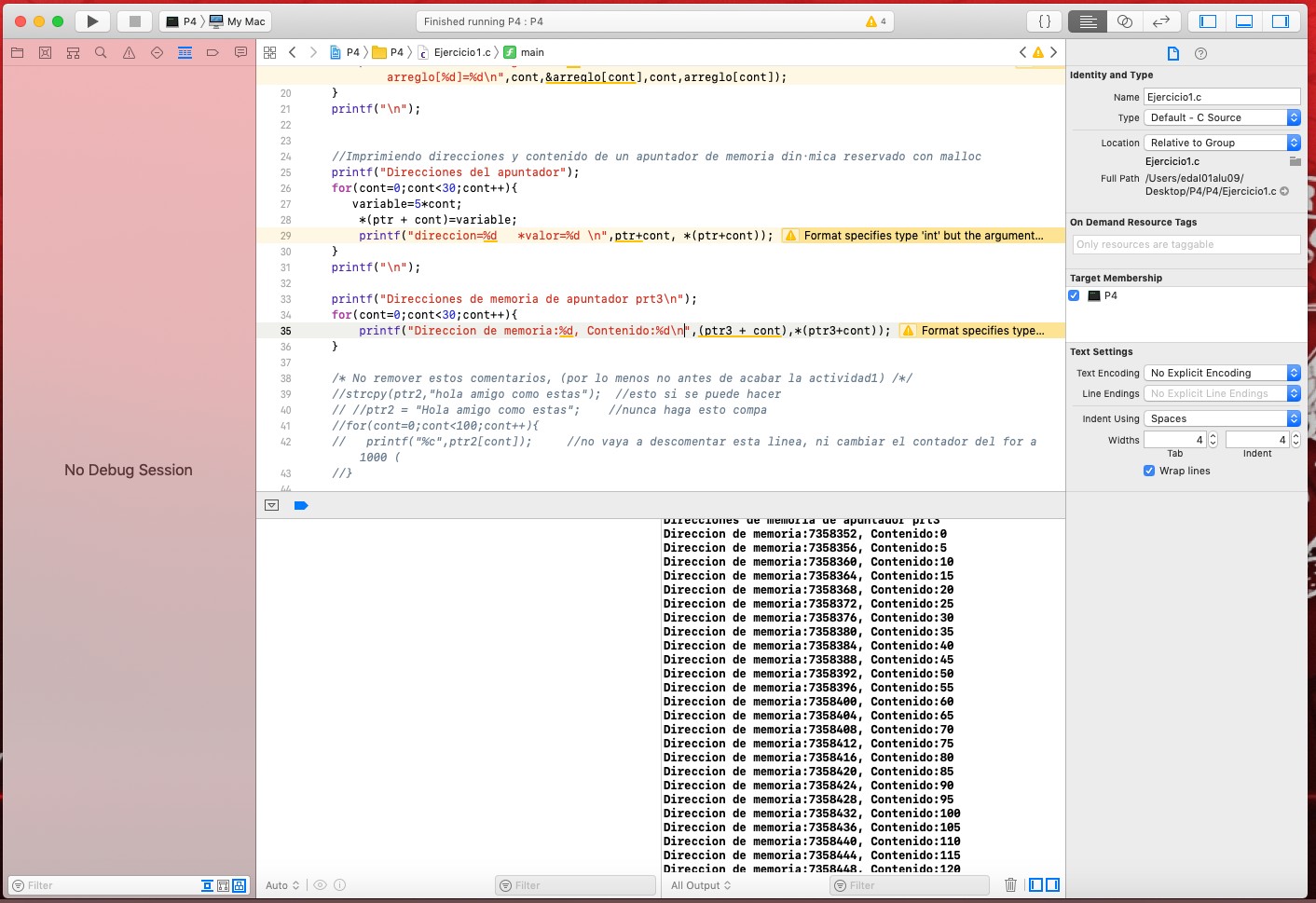
Al utilizar la función calloc, lo que sucede es que esta limpia las direcciones de memoria, por ello cuando se imprime el contenido de las direcciones los valores de estas son ceros, y las direcciones de memoria c¡son contiguas con diferencia de cuatro bytes por dato.



Cuarto inciso.

El contenido que se imprimió de ptr3 es el mismo que el de ptr cuando se le daban los valores de los múltiplos de 5.

La diferencia radica en que posición de la función se haga la asignación de memoria porque, cuando se asigna mas memoria aun apuntador que previamente tenia algo guardado este conserva el contenido pero si se desea asignar nuevos valores por medio del teclado este no lo permite, es algo raro. En el programa como le asigne memoria al mismo apuntador estando este vacío no había problema.



Ejercicio 2.

El problemas al momento de compilar pueden ser dos cosas, el compilador que no reconoce la función free para liberar memoria y la otra problemática es que si reconoce el archivo Alumno.h, pero no termina de reconocer las instrucciones que tienen que ver con strcpy eso me pasaba en la practica 3 donde los strcpy los detectaban como inválidos, pero eso tiene solución.

Ademas intente meter el programa Alumno.h al programa2 pero aun así no lograba compilar el programa por lo que no había salidas.

