|  | **Caratula para entrega de Prácticas** |
|---|---|
| Facultad de Ingeniería | Laboratorio de docencia |

# Laboratorios de computación salas A y B

Profesor:_____Ing. Marco Antonio Martinez Quintana____

Asignatura:__Estructura de Datos y Algoritmos I (1227)_

Grupo:_____17_____

No. de Práctica(s):_____4°_____

Integrante(s):_____Avila Laguna Ricardo_____

No. de Equipo de
cómputo empleado:_____10_____

No. Lista o Brigada:_____6_____

Semestre:_____2°_____

Fecha de entrega:____10 de Marzo del 2020_____

Observaciones:_____

_____

CALIFICACIÓN:_____

## 1° Objetivos

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

## 2° 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.

## 3° Desarrollo y Resultados

**Actividades:**

- **Utilizar funciones para reservar memoria dinámica en lenguaje C.**
- **Almacenar información en los elementos reservados con memoria dinámica.**

**Código:**

```c
//MALLOC

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;
}
```

Para el primer programa que es MALLOC tenemos un arreglo de tipo apuntador a entero,con 2 variables enteras aparte los cuales serviran para saber el tamaño de numeros del conjunto que imprimira la sucecion de 1 hasta el numero dado, para lo cual cuando el vector a sido reservado nos lo dira y al fianl se liberaran los apuntadores con la funcion free.

```c
//CALLOC

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;
}
```

En el segundo programa tenemos a Calloc el cual igual va inicializarse como apuntador a entero, junto con otras 2 variables del tipo entero y con ello pide al usuario la cantidad de elementos del arreglo para poder reservar memoria y a cada dato le asigna un numero para mostrarlo al usuario, despues de ello libera la memoria. Lo que cabe resaltar de calloc es su inicialización en 0 de cada dirección a la que apunta, en diferencia con malloc.

```c
#include <stdio.h>
#include <stdlib.h>

//REALLOC

int main (){
        int *arreglo, *arreglo2, num, cont;
        printf("¿Cuántos 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));
                }
```

Para el tercer y último programa de la función REALLOC tenemos 2 apuntadores a enteros y 2 variables enteras las cuales cuando inicia el programa le piden al usuario el número de elementos de un arreglo para reservar su memoria con malloc, después con un ciclo for tenemo que digitar el valor de cada dato del arreglo.

```c
                printf("Vector insertado:\n\t[");
                for (cont=0 ; cont < num ; cont++){
                        printf("\t%d",*(arreglo+cont));
                }
                printf("\t]\n");
                printf("Aumentando el tamaño del conjunto al doble.\n");
                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));
                        }
```

Despues el programa nos imprimira todos los datos de cada valor recervado que le dimos anteriormente, para después aumentar el tamaño del arreglo al doble con la función realloc y volvernos a pedir valores para los demas espacios reservados e imprimirnos todos los datos al final.

```
25                      arreglo = arreglo2;
26                      for (; cont < num ; cont++){
27                              printf("Inserte el elemento %d del conjunto.\n",cont+1);
28                              scanf("%d",(arreglo2+cont));
29                      }
30                      printf("Vector insertado:\n\t[");
31                      for (cont=0 ; cont < num ; cont++){
32                              printf("\t%d",*(arreglo2+cont));
33                      }
34                      printf("\t]\n");
35              }
36              free (arreglo);
37      }
38      return 0;
39  }
```

Por ultimo liberamos la memoria con la función free para finalizar con un return 0.

## 4° Conclusiones

Avila Laguna Ricardo :

Los objetivos se cumplieron ya que utilizamos funciones programadas en lenguaje C para poder reservar memoria, de los cuales vimos 2 tipos diferentes de funciones Malloc y Calloc en la cual ambos reservan una memoria necesaria para un arreglo con apuntadores pero su diferencia es que Calloc inicializa en 0. Mientras que la 3 función fue Realloc la cual redimenciona el arreglo llegando a ser más grande o más pequeño según lo necesitemos, gracias a todas estas funciones podemos armar algo más grande y eficiente llamado MEMORIA DINAMICA.

## Bibliografía


http://lcp02.fi-b.unam.mx/