



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* Marco Antonio Martínez Quintana

*Asignatura:* Estructura de Datos y Algoritmos I

*Grupo:* 17

*No de Práctica(s):* 02

*Integrante(s):* Pacheco Barbosa Angel David

*No. de Equipo de  
cómputo* 24

*No. de Lista o*

*Semestre:* 2020-2

*Fecha de entrega:* 16 de febrero

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

### Objetivo

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

### Introducción

La memoria dinámica es el espacio de almacenamiento que se reserva/usa en tiempo de ejecución, ya 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 y Resultados.

Códigos: (Realizado en Editor de Textos de Ubuntu)

#### **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;
}
```

```
angeldpb2@angeldpb2-ZX4970:~$ cd 'EDA I'
angeldpb2@angeldpb2-ZX4970:~/EDA I$ gcc malloc.c -o malloc
angeldpb2@angeldpb2-ZX4970:~/EDA I$ ./malloc
¿Cuantos elementos tiene el conjunto?
6
Vector reservado:
      [      0      0      0      0      0      0      0      ]
Se libera el espacio reservado.
angeldpb2@angeldpb2-ZX4970:~/EDA I$
```

## 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;
}
```

```
angeldpb2@angeldpb2-ZX4970:~/EDA I$ gcc calloc.c -o calloc
angeldpb2@angeldpb2-ZX4970:~/EDA I$ ./calloc
¿Cuantos elementos tiene el conjunto?
6
Vector reservado:
[      0      0      0      0      0      0      0      ]
Se libera el espacio reservado.
angeldpb2@angeldpb2-ZX4970:~/EDA I$
```

## REALLOC

---

```
#include <stdio.h>
#include <stdlib.h>
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));
        }
        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));
            }
            printf("\t]\n");
        }
        free (arreglo);
    }
    return 0;
}
```

```

angeldpb2@angeldpb2-ZX4970:~/EDA I$ ./realloc
¿Cuántos elementos tiene el conjunto?
3
Inserte el elemento 1 del conjunto.
1
Inserte el elemento 2 del conjunto.
2
Inserte el elemento 3 del conjunto.
3
Vector insertado:
      [      1      2      3      ]
Aumentando el tamaño del conjunto al doble.
Inserte el elemento 4 del conjunto.
4
Inserte el elemento 5 del conjunto.
5
Inserte el elemento 6 del conjunto.
6
Vector insertado:
      [      1      2      3      4      5      6      ]
angeldpb2@angeldpb2-ZX4970:~/EDA I$ █

```

### **Explicación del Código:**

El código usa el malloc, calloc y realloc para realizar (básicamente) la misma función, poder reservar memoria durante la ejecución y limpiarla después.

El malloc pone la basura que tenga en la memoria(aunque en este caso la basura era 0) y la imprime tal cual; en cambio, el calloc inicializa lo que se va a imprimir en 0.

El realloc por otra parte pide el elemento que se va a insertar en el elemento, por lo tanto imprime ese elemento. Lo que añade el código del malloc es que duplica el número de elementos que se pidió y los reserva durante la ejecución después de la primera reserva.

### **Conclusión**

La práctica me sirvió para comprender el malloc, calloc y realloc mejor, pues cuando los revisé en mi anterior curso no me quedaron claros, así que ahora aplicándolos para la codificación de un programa, me sirvió para entenderlos. La práctica fue satisfactoria y tanto el código y su ejecución se realizaron de una manera satisfactoria.

### **Bibliografía**

Guía de la Práctica 04 de Estructura de Datos y Algoritmos I.