

# Ejemplo de asignación, redimensionamiento y liberación de memoria

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define ERROR 1

struct persona
{
    char nombre[50];
    int edad;
};

int main()
{
    struct persona *p;
    int seguircargando;
    int longitudPersonas = 1;
    struct persona *aux;
    int i;
    // Creamos el array de personas
    p = malloc(sizeof(struct persona));
    if(p == NULL)
    {
        printf("\n No hay lugar en la memoria");
        return ERROR;
    }

    do
    {
        printf("\n Ingrese nombre:");
        scanf("%s", (p+longitudPersonas-1)->nombre);
        printf("\n Ingrese edad:");
        scanf("%d", &(p+longitudPersonas-1)->edad);
        printf("\n Desea cargar otra persona ingrese (1):");
        scanf("%d", &seguircargando);
        if(seguircargando ==1)
```

```
{
    // calcular el nuevo tamaño del array
    // Redimensionamos la lista
    aux = realloc(p,sizeof(struct persona));
    if( aux == NULL)
    {
        printf("\n No hay lugar en la memoria");
        return ERROR;
    }
    longitudPersonas++; //Incremento el contador de personas
    p = aux;
}
}
while(seguircargando==1);

for(i =0; i <longitudPersonas; i++)
{
    printf("\n-----\n");
    printf("\n Nombre: %s - ", (p+i)->nombre);
    printf("Edad: %d", (p+i)->edad);
}
free(p); // liberamos memoria

return 0;
}
```