

Malloc

(Muestra datos basura en la memoria o residuos de programas anteriores)

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main () {
4      int *arreglo, num, cont;
5      //Te pide valor del conjunto
6      printf(";Cuantos elementos tiene el conjunto?\n");
7      scanf("%d",&num);
8      //Se usa "Malloc"
9      //Reserva espacio de memoria
10     arreglo = (int *)malloc (num * sizeof(int));
11     //Imprime los valores de datos guardados en nuestro conjunto
12     if (arreglo!=NULL) {
13         printf("Vector reservado:\n\t[");
14         for (cont=0 ; cont<num ; cont++){
15             printf("\t%d",*(arreglo+cont));
16         }
17         printf("\t]\n");
18         printf("Se libera el espacio reservado.\n");
19         free(arreglo);
20     }
21     return 0;
22 }
```

Te pide cuantos elementos tiene el conjunto y la función malloc te muestra el valor de los datos que están guardados en ese espacio.

```
1 Cuantos elementos tiene el conjunto?
2
Vector reservado:
[      12661304      12648640      ]
Se libera el espacio reservado.

Process returned 0 (0x0)   execution time : 3.318 s
Press any key to continue.
```

```
1 Cuantos elementos tiene el conjunto?
2
Vector reservado:
[      7405760 7414288 0      ]
Se libera el espacio reservado.

Process returned 0 (0x0)   execution time : 1.992 s
Press any key to continue.
```

```
1 Cuantos elementos tiene el conjunto?
2
Vector reservado:
[      131264 139792 0      0      0      0      0      ]
Se libera el espacio reservado.

Process returned 0 (0x0)   execution time : 2.025 s
Press any key to continue.
```

Calloc (Limpia memoria)

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main () {
4      int *arreglo, num, cont;
5      //Te pide valor del conjunto
6      printf("¿Cuántos elementos tiene el conjunto?\n");
7      scanf("%d", &num);
8      //Se usa "Calloc"
9      //Reserva espacio de memoria y los deja en valor cero
10     arreglo = (int *)calloc (num, sizeof(int));
11     //Imprime el vector de memoria limpia
12     if (arreglo!=NULL) {
13         printf("Vector reservado:\n\t[");
14         for (cont=0 ; cont<num ; cont++){
15             printf("\t%d", *(arreglo+cont));
16         }
17         printf("\t]\n");
18         printf("Se libera el espacio reservado.\n");
19         free(arreglo);
20     }
21     return 0;
22 }
```

La función calloc funciona de manera similar a la función malloc pero, además de reservar memoria en tiempo real, inicializa la memoria reservada con 0.

```
¿Cuántos elementos tiene el conjunto?
2
Vector reservado:
[      0      0      ]
Se libera el espacio reservado.

Process returned 0 (0x0)   execution time : 4.021 s
Press any key to continue.
```

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

Process returned 0 (0x0)   execution time : 1.887 s
Press any key to continue.
```

```
¿Cuántos elementos tiene el conjunto?
7
Vector reservado:
[      0      0      0      0      0      0      0      ]
Se libera el espacio reservado.

Process returned 0 (0x0)   execution time : 1.835 s
Press any key to continue.
```

Realloc (Redimensiona el espacio de memoria)

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main () {
4      int *arreglo, *arreglo2, num, cont;
5      //Te pide valor del conjunto
6      printf("¿Cuántos elementos tiene el conjunto?\n");
7      scanf("%d",&num);
8      //Se usa "Malloc"
9      arreglo = (int *)malloc (num * sizeof(int));
10     if (arreglo!=NULL) {
11         for (cont=0 ; cont < num ; cont++){
12             printf("Inserte el elemento %d del conjunto.\n",cont+1);
13             scanf("%d", (arreglo+cont));
14         }
15         printf("Vector insertado:\n\t[");
16         for (cont=0 ; cont < num ; cont++){
17             printf("\t%d",*(arreglo+cont));
18         }
19         printf("\t]\n");
20         printf("Aumentando el tamaño del conjunto al doble.\n");
21         num *= 2;
22         //Se usa la funcion realloc
23         //Aumenta el tamaño de memoria reservada
24         arreglo2 = (int *)realloc (arreglo,num*sizeof(int));
25         if (arreglo2 != NULL) {
26             arreglo = arreglo2;
27             //Ingresas nuevos valores a nuestra nueva dimension de memoria
28             for (; cont < num ; cont++){
29                 printf("Inserte el elemento %d del conjunto.\n",cont+1);
30                 scanf("%d", (arreglo2+cont));
31             }
32             printf("Vector insertado:\n\t[");
33             for (cont=0 ; cont < num ; cont++){
34                 printf("\t%d",*(arreglo2+cont));
35             }
36             printf("\t]\n");
37         }
38         free (arreglo);
39     }
40     return 0;
41 }
42
```

La función `realloc` permite redimensionar el espacio asignado previamente de forma dinámica, es decir, permite aumentar el tamaño de la memoria reservada de manera dinámica. Su sintaxis es la siguiente:

Si el apuntador que se desea redimensionar tiene el valor nulo, la función actúa como la función `malloc`. Si la reasignación no se pudo realizar, la función devuelve un apuntador a nulo, dejando intacto el apuntador que se pasa como parámetro (el espacio reservado previamente).

```
¿Cuántos elementos tiene el conjunto?
4
Inserte el elemento 1 del conjunto.
1
Inserte el elemento 2 del conjunto.
2
Inserte el elemento 3 del conjunto.
3
Inserte el elemento 4 del conjunto.
4
Vector insertado:
      [      1      2      3      4      ]
Aumentando el tamaño del conjunto al doble.
Inserte el elemento 5 del conjunto.
5
Inserte el elemento 6 del conjunto.
6
Inserte el elemento 7 del conjunto.
7
Inserte el elemento 8 del conjunto.
8
Vector insertado:
      [      1      2      3      4      5      6      7      8      ]
]

Process returned 0 (0x0)   execution time : 27.876 s
Press any key to continue.
```