

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

void
main(void)
{
    char *a;

    a = CrearCaracter();
    *a = 'x';
    EscriureCaracter(*a);
    EscriureCaracter('\n');
    DestruirCaracter(a);
}
```

```
char *
CrearCharacter(void)
{
    char *c;

    c = malloc(sizeof (char));
    assert(c != NULL);
    return c;
}
```

```
void
DestruirCharacter(char *c)
{
    free(c);
}
```

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

```
#define M 10
```

```
typedef int TaulaEnter[M];
```

```
typedef struct {
    char c;
    int i;
    TaulaEnter t;
} TuplaA;
```

```
void
main(void)
{
    TuplaA *ta;
    int i;

    ta = CrearTuplaA();
    (*ta).c = 'v';
    (*ta).i = 321;
    i = 0;
    while (i < M) {
        (*ta).t[i] = 2 * i;
        i = i + 1;
    }
    EscriureTuplaA(*ta);
}
```

```
TuplaA *
CrearTuplaA(void)
{
    TuplaA *t;

    t = malloc(sizeof (TuplaA));
    assert(t != NULL);
    return t;
}
```

```
void
DestruirTuplaA(TuplaA * t)
{
    free(t);
}
```

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

#define N 10

typedef char TauCar[N];

void
main(void)
{
    TauCar *tc;

    tc = CrearTauCar();
    (*tc)[0] = 'a';
    (*tc)[5] = 'a';
}
```

```
TauCar *
CrearTauCar(void)
{
    TauCar *t;

    t = malloc(sizeof (TauCar));
    assert(t != NULL);
    return t;
}
```

```
void
DestruirTauCar(TauCar * t)
{
    free(t);
}
```

```
void
main(void)
{
    int *e;
    float *r;

    r = CrearReal();
    *r = 3.78;
    EscriureReal(*r);
    DestruirReal(r);
    e = CrearEnter();
    *e = 8999;
    EscriureReal(*r);
    DestruirEnter(e);
}
```

## Alliberament de memòria no reservada

---

```
void
main(void)
{
    int i, *ip;
    char *c;

    i = 3;
    ip = &i;
    EscriureEnter(i);
    free(ip);
    c = CrearCaracter();
    *c = 'a';
    DestruirCaracter(c);
    EscriureEnter(i);
}
```

```
void
main(void)
{
    int i, *ip;
    TauEnter t;
    int *t3;

    ip = &i;
    i = 3;
    /* i = ?, *ip = ? */
    *ip = 2;
    /* i = ?, *ip = ? */

    t[0] = 5;
    t[3] = 10;
```

```
void
main(void)
{
    int *p1, *p2;

    p1 = CrearEnter();
    *p1 = 2;
    p2 = CrearEnter();
    *p2 = 3;
    /* p1 = ?, p2 = ? */
    *p1 = *p2;
    /* p1 = ?, p2 = ? */
    p1 = p2;
    /* p1 = ?, p2 = ? */
    DestruirEnter(p1);
```