

```

#include <iostream> // Biblioteca para E/S estándar
#include <cstdio> // Biblioteca para operaciones de E/S
#include <cstdlib> // Biblioteca para gestión de memoria dinámica

#define MAXTALLA 20 // Define macro llamada con valor de 20

struct MiVector{
    int * buffer;
    int talla;
};

void swap(int * ptr1, int * ptr2){ // Intercambia los valores apuntados por dos punteros a enteros

    int temp = *ptr1;
    *ptr1 = *ptr2;
    *ptr2 = temp;
}

void ordenar(struct MiVector* v){ //Ordena el array v por método burbuja

    for (int i = v->talla -1; i>=1; i--){
        for (int j = 0; j < i; j++){
            if (v->buffer[j] > v->buffer[j+1]){
                swap(&v->buffer[j], &v->buffer[j+1]);
            }
        }
    }
}

void leer(struct MiVector* v){ //Solicita al usuario que introduzca nums, y los almacena en el vector
"buffer"
    int i;

    for (i = 0; i < v->talla; i++){
        printf("Introduce un número: ");
        scanf("%d", &v -> buffer[i]);
    }
}

void imprimir(struct MiVector* v){

    for (int i = 0; i < v->talla; i++){
        printf("%d", v->buffer[i]);
    }
    printf("\n");
}

void iniciar(struct MiVector* v, int n){
    if (n > 0){
        v->talla = n;
    }
    else{
        v->talla = MAXTALLA;
        v->buffer = (int*)malloc(v->talla * sizeof(int));
    }
}

void liberar(struct MiVector * v){
    free(v->buffer);
}

int main(){

    struct MiVector v1, v2;

    iniciar(&v1, 5);
    iniciar(&v2, 7);

    leer(&v1);
    ordenar(&v1);
    imprimir(&v1);

    leer(&v2);

```

```
ordenar(&v2);  
imprimir(&v2);
```

```
liberar(&v1);  
liberar(&v2);
```

```
return 0;
```

```
}
```