

EJERCICIO 1

```
#include <stdio.h>
#define SIZE 7
void main(void) {
    int vector[SIZE];
    int j, i, temp;
    printf("Introduce los %d valores para ordenar:\n", SIZE);
    for(i=0; i<SIZE; i++) {
        printf("%d: ", i+1);
        scanf("%d", &vector[i]);
        printf("\n");
    }
    /* se aplica el algoritmo de la burbuja */
    for(i=0; i<(SIZE-1); i++) {
        for (j=i+1; j<SIZE; j++) {
            if(vector[j]<vector[i]) {
                temp=vector[j];
                vector[j]=vector[i];
                vector[i]=temp;
            }
        }
    }
    printf("El vector ordenado es:\n");
    for(i=0; i<SIZE ; i++) {
        printf("%d ", vector[i]);
    }
    printf("\n");
}
```

EJERCICIO 2

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

void OrdenaBurbuja (float v[], int n)
{
    int t, h, e;
    for (h=1; h<n; h++)
    {
        for(e=0; e<n;e++)
        {
```

```

if (v[e]>v[e+1])
{
v[e+1] = t;
v[e] = v[e+1];
t= v[e];
}
}
}
}

```

```

void ImprimirVector(float v[], int n)
{
int i;
for (i=0; i<n; i++)
{
printf("%f ", v[i]);
}
}

```

```

int main(int argc, char *argv[])
{
if(strcmp(argv[1], "burbuja")==0)
{
int n, i, m=2, o;
float *v;
{
n=argc-2;
v =(float *)malloc(n*sizeof(float));

```

```
for (i=0; i<n; i++)  
{  
v[i] = atof(argv[i+2]);  
}  
  
OrdenaBurbuja(v, n);  
ImprimirVector(v, n);  
}  
  
system("PAUSE");  
return 0;  
}
```