

Código en C

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

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

```
void llenarArreglo(int* arreglo, int tamanio);
```

```
void mostrarArreglo(int* arreglo,int tamanio);
```

```
void ordenarBurbuja(int* arreglo, int tamanio);
```

```
int main(int argc, char* argv[]){
```

```
    int n;
```

```
    srand(getpid());
```

```
    if (argc!=2){
```

```
        printf("\nIndique el tamaño como argumento\n");
```

```
        exit(1);
```

```
    }else{
```

```
        n=atoi(argv[1]);
```

```
    }
```

```
    int* A=(int*)malloc(n*sizeof(int));
```

```
    llenarArreglo(A,n);
```

```
    printf("Numeros desordenados: ");
```

```
    mostrarArreglo(A,n);
```

```
    ordenarBurbuja(A,n);
```

```
    printf("\nNumeros ordenados: ");
```

```
    mostrarArreglo(A,n);
```

```
    return 0;
```

```
}
```

```
void llenarArreglo(int* arreglo, int tamanio){
```

```
    int i;
```

```
    for(i=0;i<tamanio;i++){
```

```
        arreglo[i]=rand();
```

```

    }
}

void mostrarArreglo(int* arreglo,int tamano){
    int i;

    for(i=0;i<tamano;i++){
        printf("%d ",arreglo[i]);
    }
}

void ordenarBurbuja(int* arreglo, int tamano){
    int i,j,aux;

    for(i=0; i<tamano-1; i++){
        for(j=0; j<tamano-i-1; j++){
            if(arreglo[j]>arreglo[j+1]){
                aux = arreglo[j];
                arreglo[j] = arreglo[j+1];
                arreglo[j+1] = aux;
            }
        }
    }
}

```

Código en ensamblador

```

.file    "burbuja.c"
.text
.section     .rodata
.align 8

.LC0:
.string "\nIndique el tamano como argumento"

.LC1:
.string "Numeros desordenados: "

.LC2:

```

```

.string "\nNumeros ordenados: "
.text
.globl main
.type main, @function
main:
.LFB6:
.cfi_startproc
endbr64
pushq %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq %rsp, %rbp
.cfi_def_cfa_register 6
subq $32, %rsp
movl %edi, -20(%rbp)
movq %rsi, -32(%rbp)
movl $0, %eax
call getpid@PLT
movl %eax, %edi
call srand@PLT
cmpl $2, -20(%rbp)
je .L2
leaq .LC0(%rip), %rdi
call puts@PLT
movl $1, %edi
call exit@PLT
.L2:
movq -32(%rbp), %rax
addq $8, %rax
movq (%rax), %rax
movq %rax, %rdi
call atoi@PLT
movl %eax, -12(%rbp)
movl -12(%rbp), %eax
cltq

```

```
salq    $2, %rax
movq    %rax, %rdi
call    malloc@PLT
movq    %rax, -8(%rbp)
movl    -12(%rbp), %edx
movq    -8(%rbp), %rax
movl    %edx, %esi
movq    %rax, %rdi
call    llenarArreglo
leaq    .LC1(%rip), %rdi
movl    $0, %eax
call    printf@PLT
movl    -12(%rbp), %edx
movq    -8(%rbp), %rax
movl    %edx, %esi
movq    %rax, %rdi
call    mostrarArreglo
movl    -12(%rbp), %edx
movq    -8(%rbp), %rax
movl    %edx, %esi
movq    %rax, %rdi
call    ordenarBurbuja
leaq    .LC2(%rip), %rdi
movl    $0, %eax
call    printf@PLT
movl    -12(%rbp), %edx
movq    -8(%rbp), %rax
movl    %edx, %esi
movq    %rax, %rdi
call    mostrarArreglo
movl    $0, %eax
leave
.cfi_def_cfa 7, 8
ret
.cfi_endproc
```

.LFE6:

```
.size    main, .-main
.globl   llenarArreglo
.type    llenarArreglo, @function
```

llenarArreglo:

.LFB7:

```
.cfi_startproc
endbr64
pushq   %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq    %rsp, %rbp
.cfi_def_cfa_register 6
pushq   %rbx
subq    $40, %rsp
.cfi_offset 3, -24
movq    %rdi, -40(%rbp)
movl    %esi, -44(%rbp)
movl    $0, -20(%rbp)
jmp     .L5
```

.L6:

```
movl    -20(%rbp), %eax
cltq
leaq    0(%rax,4), %rdx
movq    -40(%rbp), %rax
leaq    (%rdx,%rax), %rbx
call    rand@PLT
movl    %eax, (%rbx)
addl    $1, -20(%rbp)
```

.L5:

```
movl    -20(%rbp), %eax
cmpl    -44(%rbp), %eax
jl      .L6
nop
nop
```

```

    addq $40, %rsp
    popq %rbx
    popq %rbp
    .cfi_def_cfa 7, 8
    ret
    .cfi_endproc
.LFE7:
    .size    llenarArreglo, .-llenarArreglo
    .section      .rodata
.LC3:
    .string "%d "
    .text
    .globl  mostrarArreglo
    .type   mostrarArreglo, @function
mostrarArreglo:
.LFB8:
    .cfi_startproc
    endbr64
    pushq %rbp
    .cfi_def_cfa_offset 16
    .cfi_offset 6, -16
    movq %rsp, %rbp
    .cfi_def_cfa_register 6
    subq $32, %rsp
    movq %rdi, -24(%rbp)
    movl %esi, -28(%rbp)
    movl $0, -4(%rbp)
    jmp  .L8
.L9:
    movl -4(%rbp), %eax
    cltq
    leaq 0(%rax,4), %rdx
    movq -24(%rbp), %rax
    addq %rdx, %rax
    movl (%rax), %eax

```

```

    movl    %eax, %esi
    leaq    .LC3(%rip), %rdi
    movl    $0, %eax
    call    printf@PLT
    addl    $1, -4(%rbp)
.L8:
    movl    -4(%rbp), %eax
    cmpl    -28(%rbp), %eax
    jl      .L9
    nop
    nop
    leave
    .cfi_def_cfa 7, 8
    ret
    .cfi_endproc
.LFE8:
    .size    mostrarArreglo, .-mostrarArreglo
    .globl   ordenarBurbuja
    .type    ordenarBurbuja, @function
ordenarBurbuja:
.LFB9:
    .cfi_startproc
    endbr64
    pushq   %rbp
    .cfi_def_cfa_offset 16
    .cfi_offset 6, -16
    movq    %rsp, %rbp
    .cfi_def_cfa_register 6
    movq    %rdi, -24(%rbp)
    movl    %esi, -28(%rbp)
    movl    $0, -12(%rbp)
    jmp     .L11
.L15:
    movl    $0, -8(%rbp)
    jmp     .L12

```

.L14:

```
    movl    -8(%rbp), %eax
    cltq
    leaq    0(%rax,4), %rdx
    movq    -24(%rbp), %rax
    addq    %rdx, %rax
    movl    (%rax), %edx
    movl    -8(%rbp), %eax
    cltq
    addq    $1, %rax
    leaq    0(%rax,4), %rcx
    movq    -24(%rbp), %rax
    addq    %rcx, %rax
    movl    (%rax), %eax
    cmpl    %eax, %edx
    jle     .L13
    movl    -8(%rbp), %eax
    cltq
    leaq    0(%rax,4), %rdx
    movq    -24(%rbp), %rax
    addq    %rdx, %rax
    movl    (%rax), %eax
    movl    %eax, -4(%rbp)
    movl    -8(%rbp), %eax
    cltq
    addq    $1, %rax
    leaq    0(%rax,4), %rdx
    movq    -24(%rbp), %rax
    addq    %rdx, %rax
    movl    -8(%rbp), %edx
    movslq    %edx, %rdx
    leaq    0(%rdx,4), %rcx
    movq    -24(%rbp), %rdx
    addq    %rcx, %rdx
    movl    (%rax), %eax
```



```

    movl    %eax, (%rdx)
    movl    -8(%rbp), %eax
    cltq
    addq    $1, %rax
    leaq    0(%rax,4), %rdx
    movq    -24(%rbp), %rax
    addq    %rax, %rdx
    movl    -4(%rbp), %eax
    movl    %eax, (%rdx)
.L13:
    addl    $1, -8(%rbp)
.L12:
    movl    -28(%rbp), %eax
    subl    -12(%rbp), %eax
    subl    $1, %eax
    cmpl    %eax, -8(%rbp)
    jl      .L14
    addl    $1, -12(%rbp)
.L11:
    movl    -28(%rbp), %eax
    subl    $1, %eax
    cmpl    %eax, -12(%rbp)
    jl      .L15
    nop
    nop
    popq    %rbp
    .cfi_def_cfa 7, 8
    ret
    .cfi_endproc
.LFE9:
    .size    ordenarBurbuja, .-ordenarBurbuja
    .ident   "GCC: (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0"
    .section .note.GNU-stack,"",@progbits
    .section .note.gnu.property,"a"
    .align 8

```

```
    .long 1f - 0f
    .long 4f - 1f
    .long 5
0:
    .string "GNU"
1:
    .align 8
    .long 0xc0000002
    .long 3f - 2f
2:
    .long 0x3
3:
    .align 8
4:
```