



Instituto Politécnico Nacional

Escuela Superior de Computo



Practica 2.

Lenguaje ensamblador.

Nombre: Flores Castro Luis Antonio.

Arquitectura de Computadoras.

Profesora: Vega García Nayeli.

Algoritmo de Ordenamiento Burbuja (Código en C).

```
1. #include<stdio.h>
2. #include<stdlib.h>
3.
4. void llenararreglo();
5. void mostrararreglo(int[]);
6. void ordenarburbuja(int[]);
7. #define TAM 20
8.
9. int main(void)
10. {
11.     llenararreglo();
12.     system("pause");
13.     return 0;
14. }
15.
16. void llenararreglo()
17. {
18.     int arr[TAM];
19.     int i;
20.     //llenado de arreglo
21.     for(i=0; i<20; i++)
22.         arr[i]=rand()%100;
23.
24.     mostrararreglo(arr);
25.     printf("\n");
26.     ordenarburbuja(arr);
27. }
28.
29. void mostrararreglo(int arr[])
30. {
31.     int i;
32.     for(i=0; i<20; i++)
33.         printf("\n%d",arr[i]);
34. }
35.
36. void ordenarburbuja(int arr[])
37. {
38.     int j,cambio;
39.     int flag=19;
40.     while(flag>0)
41.     {
42.         cambio=0;
43.         for(j=0; j<19; j++)
44.         {
45.             if(arr[j]>arr[j+1])
46.             {
47.                 int aux=arr[j];
48.                 arr[j]=arr[j+1];
```

```
49.                                     arr[j+1]=aux;
50.                                     cambio=j;
51.                                     }
52.                                     flag=cambio;
53.                                 }
54.    }
55.    mostrararreglo(arr);
56.    printf("\n");
57. }
```

Algoritmo de Ordenamiento Burbuja (Código en ensamblador).

```
1. .file    "Burbuja.c"
2.         .text
3.         .section        .rodata
4. .LC0:
5.         .string "pause"
6.         .text
7.         .globl  main
8.         .type   main, @function
9. main:
10.      .LFB6:
11.          .cfi_startproc
12.          endbr64
13.          pushq   %rbp
14.          .cfi_def_cfa_offset 16
15.          .cfi_offset 6, -16
16.          movq    %rsp, %rbp
17.          .cfi_def_cfa_register 6
18.          movl    $0, %eax
19.          call    llenararreglo
20.          leaq    .LC0(%rip), %rdi
21.          call    system@PLT
22.          movl    $0, %eax
23.          popq    %rbp
24.          .cfi_def_cfa 7, 8
25.          ret
26.          .cfi_endproc
27.      .LFE6:
28.          .size    main, .-main
29.          .globl  llenararreglo
30.          .type    llenararreglo, @function
31. llenararreglo:
32.      .LFB7:
33.          .cfi_startproc
34.          endbr64
35.          pushq   %rbp
36.          .cfi_def_cfa_offset 16
37.          .cfi_offset 6, -16
38.          movq    %rsp, %rbp
39.          .cfi_def_cfa_register 6
40.          subq    $112, %rsp
41.          movq    %fs:40, %rax
42.          movq    %rax, -8(%rbp)
43.          xorl    %eax, %eax
44.          movl    $0, -100(%rbp)
45.          jmp     .L4
46.      .L5:
47.          call    rand@PLT
48.          movslq  %eax, %rdx
```

```

49.      imulq    $1374389535, %rdx, %rdx
50.      shrq    $32, %rdx
51.      sarl    $5, %edx
52.      movl    %eax, %ecx
53.      sarl    $31, %ecx
54.      subl    %ecx, %edx
55.      imull    $100, %edx, %ecx
56.      subl    %ecx, %eax
57.      movl    %eax, %edx
58.      movl    -100(%rbp), %eax
59.      cltq
60.      movl    %edx, -96(%rbp,%rax,4)
61.      addl    $1, -100(%rbp)
62.      .L4:
63.      cmpl    $19, -100(%rbp)
64.      jle     .L5
65.      leaq    -96(%rbp), %rax
66.      movq    %rax, %rdi
67.      call    mostrararreglo
68.      movl    $10, %edi
69.      call    putchar@PLT
70.      leaq    -96(%rbp), %rax
71.      movq    %rax, %rdi
72.      call    ordenarburbuja
73.      nop
74.      movq    -8(%rbp), %rax
75.      subq    %fs:40, %rax
76.      je     .L6
77.      call    __stack_chk_fail@PLT
78.      .L6:
79.      leave
80.      .cfi_def_cfa 7, 8
81.      ret
82.      .cfi_endproc
83.      .LFE7:
84.      .size    llenararreglo, .-llenararreglo
85.      .section          .rodata
86.      .LC1:
87.      .string  "\n%d"
88.      .text
89.      .globl  mostrararreglo
90.      .type   mostrararreglo, @function
91.  mostrararreglo:
92.      .LFB8:
93.      .cfi_startproc
94.      endbr64
95.      pushq   %rbp
96.      .cfi_def_cfa_offset 16
97.      .cfi_offset 6, -16
98.      movq    %rsp, %rbp
99.      .cfi_def_cfa_register 6
100.     subq    $32, %rsp

```

```

101.      movq    %rdi, -24(%rbp)
102.      movl    $0, -4(%rbp)
103.      jmp     .L8
104.  .L9:
105.      movl    -4(%rbp), %eax
106.      cltq
107.      leaq    0(,%rax,4), %rdx
108.      movq    -24(%rbp), %rax
109.      addq    %rdx, %rax
110.      movl    (%rax), %eax
111.      movl    %eax, %esi
112.      leaq    .LC1(%rip), %rdi
113.      movl    $0, %eax
114.      call    printf@PLT
115.      addl    $1, -4(%rbp)
116.  .L8:
117.      cmpl    $19, -4(%rbp)
118.      jle     .L9
119.      nop
120.      nop
121.      leave
122.      .cfi_def_cfa 7, 8
123.      ret
124.      .cfi_endproc
125.  .LFE8:
126.      .size    mostrararreglo, .-mostrararreglo
127.      .globl   ordenarburbuja
128.      .type    ordenarburbuja, @function
129. ordenarburbuja:
130.  .LFB9:
131.      .cfi_startproc
132.      endbr64
133.      pushq    %rbp
134.      .cfi_def_cfa_offset 16
135.      .cfi_offset 6, -16
136.      movq     %rsp, %rbp
137.      .cfi_def_cfa_register 6
138.      subq     $32, %rsp
139.      movq     %rdi, -24(%rbp)
140.      movl     $19, -8(%rbp)
141.      jmp      .L11
142.  .L15:
143.      movl     $0, -12(%rbp)
144.      movl     $0, -16(%rbp)
145.      jmp      .L12
146.  .L14:
147.      movl     -16(%rbp), %eax
148.      cltq
149.      leaq     0(,%rax,4), %rdx
150.      movq     -24(%rbp), %rax
151.      addq     %rdx, %rax
152.      movl     (%rax), %edx

```

```

153.      movl    -16(%rbp), %eax
154.      cltq
155.      addq    $1, %rax
156.      leaq    0(,%rax,4), %rcx
157.      movq    -24(%rbp), %rax
158.      addq    %rcx, %rax
159.      movl    (%rax), %eax
160.      cmpl    %eax, %edx
161.      jle     .L13
162.      movl    -16(%rbp), %eax
163.      cltq
164.      leaq    0(,%rax,4), %rdx
165.      movq    -24(%rbp), %rax
166.      addq    %rdx, %rax
167.      movl    (%rax), %eax
168.      movl    %eax, -4(%rbp)
169.      movl    -16(%rbp), %eax
170.      cltq
171.      addq    $1, %rax
172.      leaq    0(,%rax,4), %rdx
173.      movq    -24(%rbp), %rax
174.      addq    %rdx, %rax
175.      movl    -16(%rbp), %edx
176.      movslq  %edx, %rdx
177.      leaq    0(,%rdx,4), %rcx
178.      movq    -24(%rbp), %rdx
179.      addq    %rcx, %rdx
180.      movl    (%rax), %eax
181.      movl    %eax, (%rdx)
182.      movl    -16(%rbp), %eax
183.      cltq
184.      addq    $1, %rax
185.      leaq    0(,%rax,4), %rdx
186.      movq    -24(%rbp), %rax
187.      addq    %rax, %rdx
188.      movl    -4(%rbp), %eax
189.      movl    %eax, (%rdx)
190.      movl    -16(%rbp), %eax
191.      movl    %eax, -12(%rbp)
192.      .L13:
193.      movl    -12(%rbp), %eax
194.      movl    %eax, -8(%rbp)
195.      addl    $1, -16(%rbp)
196.      .L12:
197.      cmpl    $18, -16(%rbp)
198.      jle     .L14
199.      .L11:
200.      cmpl    $0, -8(%rbp)
201.      jg      .L15
202.      movq    -24(%rbp), %rax
203.      movq    %rax, %rdi
204.      call    mostrararreglo

```

```

205.      movl    $10, %edi
206.      call    putchar@PLT
207.      nop
208.      leave
209.      .cfi_def_cfa 7, 8
210.      ret
211.      .cfi_endproc
212.  .LFE9:
213.      .size    ordenarburbuja, .-ordenarburbuja
214.      .ident   "GCC: (Ubuntu 10.3.0-1ubuntu1) 10.3.0"
215.      .section .note.GNU-stack,"",@progbits
216.      .section .note.gnu.property,"a"
217.      .align 8
218.      .long    1f - 0f
219.      .long    4f - 1f
220.      .long    5
221.  0:
222.      .string  "GNU"
223.  1:
224.      .align 8
225.      .long    0xc0000002
226.      .long    3f - 2f
227.  2:
228.      .long    0x3
229.  3:
230.      .align 8
231.  4:

```