Source File: `lab33.asm`
Input: Standard Input
Output: Standard Output
Value: 4

Write a program that will implement a bubble sort for an array of signed 32-bit integers. You are to constrct three functions for this assignment. The first will fill an array of signed 32-bit integers. The second will implement a bubble sort, sorting the elements in ascending order. The third will display the elements of the array. A description of the functions as well as client code for testing your implementation is shown in Figure 1, and a sample execution sequence is shown in Figure 2. To use the `Makefile` as distributed in class, add a target of `lab33` to `targets2AsmFiles`.

```
1    [list -]
2    %INCLUDE "Along32.inc"
3    %INCLUDE "Macros_Along.inc"
4    [list +]
5
6    ;-----------------------------------------------------------------------
7    extern  FillArray
8    ; HLL prototype: void FillArray(int *array, int n);
9    ; Reads n signed 32-bit integers and stores them in array.
10   ; Implements the following loop:
11   ;       for (i = 0; i < n; i++)
12   ;       {
13   ;          Read a signed 32-bit integer from standard input
14   ;          Assign the acquired integer to array[i]
15   ;       }
16   ; Receives: ESI = starting offset of array
17   ;           ECX = # of elements in array
18   ; Returns:  nothing
19   ;-----------------------------------------------------------------------
20
21   ;-----------------------------------------------------------------------
22   extern  BubbleSort
23   ; HLL prototype: void BubbleSort(int *array, int n);
24   ; Implements the following code:
25   ;       for (i = 0; i < n - 1; ++i)
26   ;       {
27   ;         for (j = 0; j < n - i - 1; ++j)
28   ;         {
29   ;           if (array[j] > array[j+1])
30   ;             swap(array[j], array[j+1])
31   ;         }
32   ;         DisplayArray(array, n)
33   ;       }
34   ; Receives: ESI = starting offset of array
35   ;           ECX = # of elements in array
36   ; Returns:  nothing
37   ;-----------------------------------------------------------------------
38
```

Figure 1. `/usr/local/3304/src/lab33main.asm` (Part 1 of 2)

```
39   ;------------------------------------------------------------------------
40   extern  DisplayArray
41   ; HLL prototype: void DisplayArray(int *array, int n);
42   ; Implements the following loop:
43   ;       for (i = 0; i < n; ++i)
44   ;          print array[i]
45   ; The elements are displayed on a single line, with elements being
46   ; separated by commas.
47   ; Receives: ESI = starting offset of array
48   ;           ECX = # of elements in array
49   ; Returns:  nothing
50   ;------------------------------------------------------------------------
51
52   SECTION .data
53   array   times   1024 dd 0
54
55   SECTION .bss
56   n       resd    1
57
58   SECTION .text
59           global  _start
60   _start:
61           call    ReadDec                 ; read an unsigned integer
62           mov     [n],eax                 ; move the integer to n
63
64           mov     esi,array
65           mov     ecx,[n]
66           call    FillArray
67
68           mov     esi,array
69           mov     ecx,[n]
70           call    DisplayArray
71
72           mov     esi,array
73           mov     ecx,[n]
74           call    BubbleSort
75
76           Exit    {0}
```

Figure 1. /usr/local/3304/src/lab33main.asm (Part 2 of 2)

```
 1   newuser@csunix ~/3304/33> cp /usr/local/3304/data/33/* .
 2   newuser@csunix ~/3304/33> cp /usr/local/3304/src/Makefile .
 3   newuser@csunix ~/3304/33> cp /usr/local/3304/src/lab33main.asm .
 4   newuser@csunix ~/3304/33> touch lab33.asm
 5   newuser@csunix ~/3304/33> make lab33
 6   nasm -f elf32  -l lab33main.lst -o lab33main.o lab33main.asm -I/usr/local/3304/include/ -I.
 7   nasm -f elf32  -l lab33.lst -o lab33.o lab33.asm -I/usr/local/3304/include/ -I.
 8   ld -m elf_i386 --dynamic-linker /lib/ld-linux.so.2 -o lab33 lab33main.o lab33.o \
 9   /usr/local/3304/src/Along32.o -lc
10   newuser@csunix ~/3304/33> ../irvine_test.sh lab33 01.dat
11   +10,+9,+8,+7,+6,+5,+4,+3,+2,+1
12   +9,+8,+7,+6,+5,+4,+3,+2,+1,+10
13   +8,+7,+6,+5,+4,+3,+2,+1,+9,+10
14   +7,+6,+5,+4,+3,+2,+1,+8,+9,+10
15   +6,+5,+4,+3,+2,+1,+7,+8,+9,+10
16   +5,+4,+3,+2,+1,+6,+7,+8,+9,+10
17   +4,+3,+2,+1,+5,+6,+7,+8,+9,+10
18   +3,+2,+1,+4,+5,+6,+7,+8,+9,+10
19   +2,+1,+3,+4,+5,+6,+7,+8,+9,+10
20   +1,+2,+3,+4,+5,+6,+7,+8,+9,+10
21   newuser@csunix ~/3304/33> ../irvine_test.sh lab33 01.dat > my.out
22   newuser@csunix ~/3304/33> diff 01.out my.out
23   newuser@csunix ~/3304/33> ../irvine_test.sh lab33 02.dat > my.out
24   newuser@csunix ~/3304/33> diff 02.out my.out
25   newuser@csunix ~/3304/33> ../irvine_test.sh lab33 03.dat > my.out
26   newuser@csunix ~/3304/33> diff 03.out my.out
27   newuser@csunix ~/3304/33>
```

Figure 2. Commands to Assemble, Link, & Run Lab 33