0 9 / 0 4 / 2 0 2 4

SORT

Prepared For :

Dr.lamiaa

Problem Description:

The provided assembly code implements the bubble sort algorithm to sort an array of integers in ascending order. The program prompts the user to enter the number of integers they want to sort, reads the integers, sorts them, and then prints the sorted array.

SAMPLE INPUT:

```
ENTER THE NUMBER
OF INPUTS: 5
ENTER THE INPUT:
45
12
67
23
1
```

SAMPLE OUTPUT:

```
THE SORTED ARRAY IS:
1 12 23 45 67
```

Updates Made to the Example:

1. **Initialization of Array:**

- The assembly code initializes an array named "array" with a size of 1000. However, only the first 5 elements are considered for sorting, as indicated by the size variable initialized to 5.

2. **Input Handling:**

- The program prompts the user to input the number of integers they wish to sort. It then enters a loop to read each integer input from the user and stores them sequentially in the array.

3. **Sorting Algorithm:**

- The sorting algorithm employed is bubble sort, implemented within the sort procedure. This algorithm iterates through the array multiple times, comparing adjacent elements and swapping them if they are in the wrong order. This process continues until the array is fully sorted.

4. **Printing Sorted Array:**

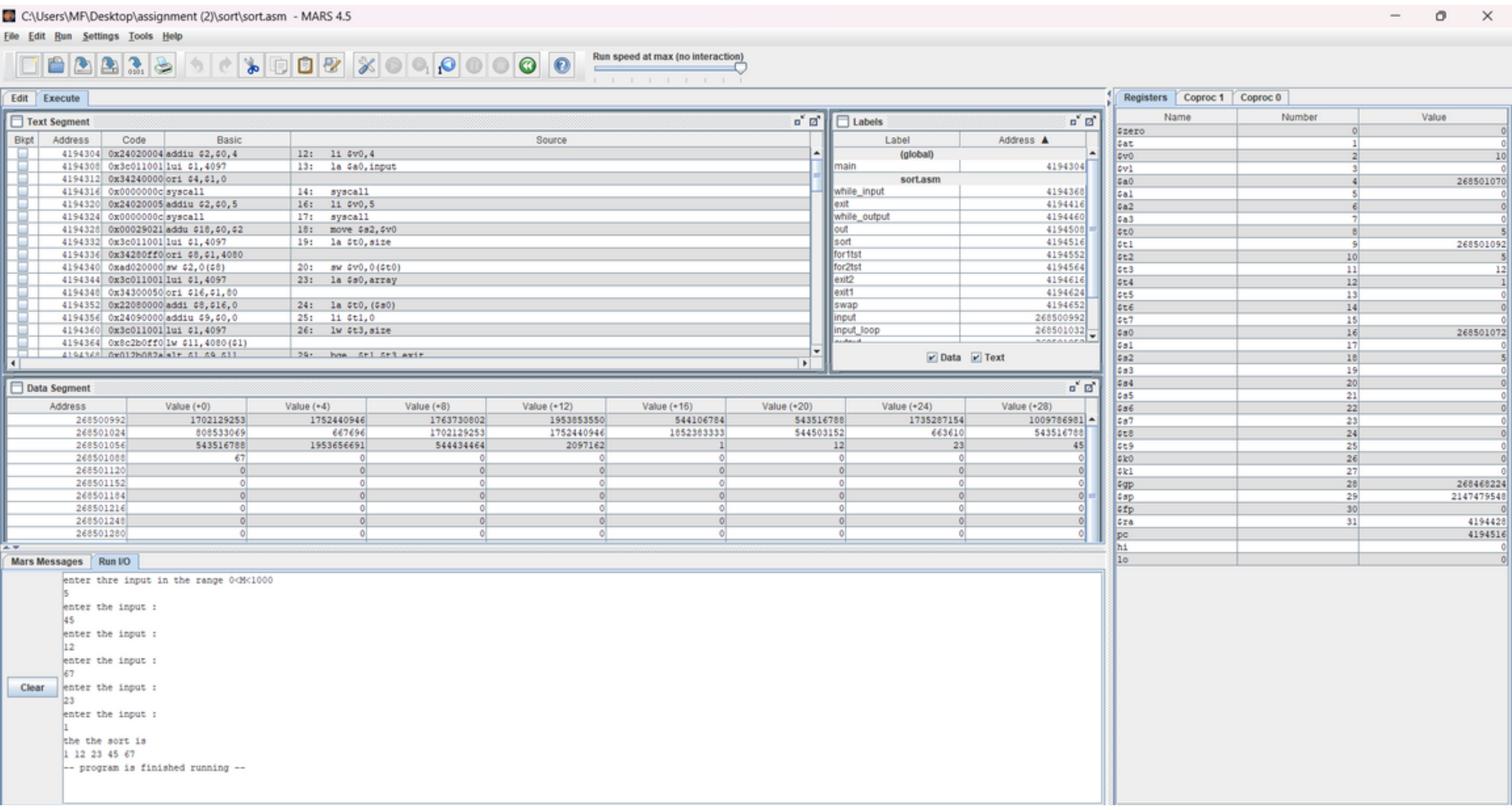
- After sorting, the program prints the sorted array. It traverses the array and prints each element separated by spaces.

5. **Exit:**

- Finally, the program exits gracefully after completing the sorting and printing tasks.

Functionality Testing:

- To ensure correctness, the program can be tested with various inputs, including edge cases such as an empty array, an array with a single element, or arrays with multiple elements. Additionally, large arrays can be tested to verify the scalability and efficiency of the sorting algorithm.


C:\Users\MF\Desktop\assignment (2)\sort\sort.asm - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Text Segment
Labels
Data Segment
Registers Coproc 1 Coproc 0
Mars Messages Run I/O
Clear
Data Text

Bkpt	Address	Code	Basic	Source
	4194304	0x24020004	addiu $2,$0,4	12: li $v0,4
	4194308	0x3c011001	lui $1,4097	13: la $a0,input
	4194312	0x34240000	ori $4,$1,0	
	4194316	0x0000000c	syscall	14: syscall
	4194320	0x24020005	addiu $2,$0,5	16: li $v0,5
	4194324	0x0000000c	syscall	17: syscall
	4194328	0x00029021	addu $18,$0,$2	18: move $s2,$v0
	4194332	0x3c011001	lui $1,4097	19: la $t0,size
	4194336	0x34280ff0	ori $8,$1,4080	
	4194340	0xad020000	sw $2,0($8)	20: sw $v0,0($t0)
	4194344	0x3c011001	lui $1,4097	23: la $s0,array
	4194348	0x34300050	ori $16,$1,80	
	4194352	0x22080000	addi $8,$16,0	24: la $t0,($s0)
	4194356	0x24090000	addiu $9,$0,0	25: li $t1,0
	4194360	0x3c011001	lui $1,4097	26: lw $t3,size
	4194364	0x8c2b0ff0	lw $11,4080($1)	
	4194368	0x012b082a	slt $1,$9,$11	29: bge $t1,$t3,exit

Label	Address ▲
(global)	
main	4194304
sort.asm	
while_input	4194368
exit	4194416
while_output	4194460
out	4194508
sort	4194516
for1tst	4194552
for2tst	4194564
exit2	4194616
exit1	4194624
swap	4194652
input	268500992
input_loop	268501032

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)
268500992	1702129253	1752440946	1763730802	1953853550	544106784	543516788	1735287154	1009786981
268501024	808533069	667696	1702129253	1752440946	1852383333	544503152	663610	543516788
268501056	543516788	1953656691	544434464	2097162	1	12	23	45
268501088	67	0	0	0	0	0	0	0
268501120	0	0	0	0	0	0	0	0
268501152	0	0	0	0	0	0	0	0
268501184	0	0	0	0	0	0	0	0
268501216	0	0	0	0	0	0	0	0
268501248	0	0	0	0	0	0	0	0
268501280	0	0	0	0	0	0	0	0

Name	Number	Value
$zero	0	0
$at	1	0
$v0	2	10
$v1	3	0
$a0	4	268501070
$a1	5	0
$a2	6	0
$a3	7	0
$t0	8	5
$t1	9	268501092
$t2	10	5
$t3	11	12
$t4	12	1
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	268501072
$s1	17	0
$s2	18	5
$s3	19	0
$s4	20	0
$s5	21	0
$s6	22	0
$s7	23	0
$t8	24	0
$t9	25	0
$k0	26	0
$k1	27	0
$gp	28	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	4194428
pc		4194516
hi		0
lo		0

```
enter thre input in the range 0<M<1000
5
enter the input :
45
enter the input :
12
enter the input :
67
enter the input :
23
enter the input :
1
the the sort is
1 12 23 45 67
-- program is finished running --
```