|  |  |  |
| --- | --- | --- |
| Comment/Remark | HEX | Assembly Instruction |
| R1 will have the value 4 | 2104 | ADDI R1,R0,4 |
| R2 will have the value 0 | 1000 | ADD R2,R0,R0 |
| R3 will have the value 0 | 1800 | ADD R3,R0,R0 |
| R4 will have the value 1 | 0c04 | ADDI R4,R0,1 |
| R6 will have the value of R4 - 1 | fe84 | ADDI R6,R4,-1 |
| R2 will have the first value in the array that’s in R6 | 02d0 | LW R2,0,R6 |
| R7 will have the first value in the array that’s in R4 | 0790 | LW R7,0,R4 |
| R2 is compared to R7 and if true branches to NEXT(R2<R7?) | 2754 | BLT R2,R7,NEXT |
| R2 is compared to R7 and if true branches to NEXT(R2==R7?) | 1f52 | BEQ R2,R7,NEXT |
| Store the value of R7 as the first element of the array that is in R6 | 07d1 | SW R7,0,R6 |
| Store the value of R2 as the first element of the array that is in R4 | 0291 | SW R2,0,R4 |
| Increment the value of R4 by 1 and store it in R4 itself. | 0c84 | ADDI R4,R4,1 |
| R2 has the value of subtracting the value R1 from the value of R3 | 5320 | SUB R2,R1,R3 |
| Comparing the values of register R4 and R2 and if true branches to FOR2(R4<R2?) | ba94 | BLT R4,R2,FOR2 |
| Increment the value of R3 by and store it in R3 itself. | 0b64 | ADDI R3,R3,1 |
| Comparing the values of register R3 and R1 and if true branches to FOR1(R3<R1?) | a174 | BLT R3,R1,FOR1 |
| Jump to end | 0018 | J end |

The bubble sort implemented here takes a 4-element array stored in the first 4 memory addresses. However, to allow it to take more than 4, change R1 to whatever number needed.

The result will be in the same memory addresses used as input to the program.

“Bubble Sort” instructions