**JoSDC Test benchmarks**

**Instructions:**

* For benchmark #1 make sure to fill out the “Expected Results” column.
* For benchmark #2:
  + If you implemented an out-of-order processor you are allowed to use Loop Unrolling for performance improvement.
  + It is recommended to start with size 20 and then transition to size 2000 for more comprehensive testing.
  + When submitting your work, please consider submitting the size 2000 version.
  + If your team encounters difficulties running the size 2000 version, you may submit the size 20 version along with proper justification.
* Any additional functionality or modifications made to the design after submitting Phase III report must be clearly documented.

**Benchmark #1:**

Instructions should be hand-assembled, converted to hexadecimal, and loaded into the instruction memory.

Initializing Registers (Testing I-Type ALU):

|  |  |  |
| --- | --- | --- |
| Instruction | Hexadecimal | Expected Result |
| **Lw r1,0(R0)** | **8C010000** | Initialize r1 = 0x0000f1e0 |
| ori r2 ,r0, 4 | **34020004** | r2 =**0x00000004** |
| addi r3, r0, -2 | **3003FFFE** | r3 = **0xFFFFFFFE** |
| **Lw r4,4(R0)** | **8C040004** | Initialize r4 = 0x0000001e |

Testing R-Type ALU Instructions (NO RAW hazards – NO Forwarding)

|  |  |  |
| --- | --- | --- |
| Instruction | Hexadecimal | Expected Result |
| add r5, r1, r1 | **00212820** | r5 = **0x0001E3C0** |
| sub r6, r1, r2 | **00223022** | r6 = **0x0000F1DC** |
| and r6, r3, r4 | **00643024** | r6 = **0x0000001E** |
| or r7, r1, r2 | **00223825** | r7 = **0x0000F1E4** |
| xor r5, r1, r3 | **00232826** | r5 = **0xFFFF0E1E** |
| nor r6, r1, r2 | **00223027** | r6 = **0xFFFF0E1B** |
| sll r7, r4, r2 | **00823800** | r7 = **0x000001E0** |
| srl r5, r1, r2 | **00222802** | r5 = **0x00000F1E** |

Testing RAW hazards and Forwarding

|  |  |  |
| --- | --- | --- |
| Instruction | Hexadecimal | Expected Result |
| add r5, r1, r1 | **00212820** | r5 = **0x0001E3C0** |
| sub r6, r5, r4 | **00A43022** | r6= **0x0001E3A2** |
| and r7, r5, r6 | **00A63824** | r7= **0x0001E380** |
| ori r5, r5, 0x000f | **34A5000F** | r5= **0x0001E3CF** |

Testing SW and LW

|  |  |  |
| --- | --- | --- |
| Instruction | Hexadecimal | Expected Result |
| sw r1, 0(r0) | **AC010000** | MEM[0] = **0x0000F1E0** |
| sw r4, 1(r0) | **AC040004** | MEM[1] = **0x0000001E** |
| lw r5, 0(r0) | **8C050000** | r5 = MEM[0] = **0x0000F1E0** |

Testing Load delay, stalling pipelining, and forwarding after LW

|  |  |  |
| --- | --- | --- |
| Instruction | Hexadecimal | Expected Result |
| lw r6, 1(r0) | **8C060004** | r6 = MEM[1] = **0x0000001E** |
| andi r7, r6,0x000b | **30C7000B** | r7 = **0x0000000A** |
| sw r7, 2(r0) | **AC070008** | MEM[2] = **0x0000000A** |
| lw r5, 0(r0) | **8C0A0000** | r5 = MEM[0] = **0x0000F1E0** |
| sw r5, 3(r0) | **AC05000C** | MEM[3] = **0x0000F1E0** |

Testing Branch and Jump Instructions

|  |  |  |
| --- | --- | --- |
| Instruction | Hexadecimal | Expected Result |
| beq r1, r1, P1 | **10210001** |  |
| add r5, r2, r2 | **00422820** |  |
| P1: bne r0, r1, P2 | **14010002** |  |
| add r6, r2, r2 | **00423020** |  |
| add r7, r4, r4 | **00843820** |  |
| P2: j P3 | **08000020** |  |
| add r5, r2, r2 | **00422820** |  |
| add r6, r4, r4 | **00843020** |  |
| P3: add r0, r0, r0 | **00000020** |  |

**Benchmark #2:**

1. Bubble Sort(20 elements):

Description/C Code:

* Declare an array of size 20 with values as given in the Assembly code below
* Initialize your design RAM and registers as needed
* Perform the given bubble sort algorithm such that the array elements are in ascending order

Assembly:

|  |
| --- |
| .data  array: .word 32000, 14849, 5683, 3987, 1024, 982, 401, 373, 183, 94, 68, 25, 6, -1, -39, -284, -620, -9102, -20375, -31000  .text  main:  jal sort  j end  sort:  la R16, array #This is a sudo instruction called "load address" that saves the initial address of array in register R16.  addi R8, R0, 0  addi R17, R0, 19    sortLoop1:  bgt R8, R17, sortEndloop1  addi R9, R0, 0  sub R10, R17, R9    sortLoop2:  bgt R9, R10, sortEndloop2  sll R11, R9, 2  add R12, R16, R11  lw R13, 0(R12)  lw R14, 4(R12)  bgt R13, R14, swap  j endSwap    swap:  add R15, R13, R0  add R13, R14, R0  add R14, R15, R0  sw R13, 0(R12)  sw R14, 4(R12)    endSwap:  addi R9, R9, 1  j sortLoop2    sortEndloop2:  addi R8, R8, 1  j sortLoop1    sortEndloop1:  jr R31    end:  add R0, R0, R0 |

1. Bubble Sort(2000 elements):

Description/C Code:

* Declare an array of size 2000 with values as given in the Assembly code below
* Initialize your design RAM and registers as needed
* Perform the given bubble sort algorithm such that the array elements are in ascending order

Assembly:

|  |
| --- |
| .data  array: .word 31995, 31991, 31988, 31986, 31950, 31900, 31878, 31847, 31754, 31734, 31683, 31636, 31609, 31583, 31567, 31559, 31559, 31556, 31533, 31398, 31366, 31360, 31302, 31261, 31246, 31069, 31046, 31004, 30985, 30927, 30926, 30898, 30864, 30858, 30844, 30838, 30797, 30746, 30714, 30696, 30666, 30590, 30563, 30550, 30527, 30460, 30367, 30228, 30208, 30169, 30078, 30073, 30040, 30031, 29946, 29905, 29891, 29890, 29773, 29763, 29729, 29717, 29717, 29621, 29594, 29535, 29528, 29513, 29490, 29490, 29458, 29443, 29372, 29368, 29346, 29292, 29285, 29225, 29223, 29222, 29142, 28993, 28952, 28952, 28870, 28865, 28808, 28801, 28724, 28679, 28669, 28617, 28603, 28589, 28579, 28577, 28522, 28518, 28501, 28471, 28451, 28433, 28411, 28366, 28341, 28288, 28288, 28246, 28236, 28229, 28193, 28182, 28170, 28148, 28063, 28044, 28027, 28004, 27946, 27892, 27855, 27844, 27824, 27817, 27739, 27728, 27724, 27723, 27700, 27676, 27634, 27606, 27505, 27483, 27449, 27416, 27415, 27376, 27356, 27334, 27333, 27296, 27235, 27232, 27218, 27211, 27177, 27165, 27116, 27046, 27032, 27016, 26949, 26907, 26895, 26786, 26767, 26753, 26750, 26723, 26711, 26702, 26671, 26544, 26526, 26472, 26458, 26444, 26438, 26405, 26392, 26361, 26338, 26329, 26281, 26279, 26271, 26130, 26118, 26097, 26072, 26054, 26044, 26044, 26039, 26039, 26018, 26003, 25938, 25818, 25818, 25804, 25761, 25673, 25629, 25494, 25485, 25388, 25319, 25302, 25237, 25217, 25203, 25197, 25187, 25082, 25050, 24967, 24954, 24919, 24913, 24893, 24870, 24835, 24824, 24778, 24755, 24711, 24707, 24658, 24629, 24618, 24601, 24534, 24463, 24422, 24415, 24390, 24381, 24377, 24355, 24248, 24219, 24189, 24182, 24179, 24149, 24107, 24097, 24086, 24083, 24020, 24003, 23970, 23919, 23885, 23827, 23821, 23732, 23712, 23657, 23636, 23573, 23566, 23562, 23541, 23517, 23500, 23499, 23464, 23463, 23438, 23414, 23413, 23365, 23363, 23360, 23350, 23318, 23318, 23285, 23284, 23237, 23216, 23205, 23187, 22978, 22976, 22937, 22921, 22876, 22851, 22843, 22828, 22765, 22764, 22743, 22677, 22657, 22601, 22541, 22499, 22496, 22488, 22483, 22465, 22403, 22387, 22361, 22318, 22312, 22311, 22277, 22172, 22145, 22053, 22024, 22000, 21981, 21975, 21836, 21765, 21746, 21721, 21681, 21670, 21653, 21615, 21606, 21586, 21583, 21561, 21557, 21525, 21521, 21494, 21493, 21436, 21384, 21362, 21353, 21351, 21287, 21277, 21273, 21271, 21262, 21180, 21115, 21088, 21050, 20990, 20951, 20915, 20880, 20819, 20780, 20750, 20740, 20667, 20638, 20589, 20565, 20524, 20519, 20477, 20391, 20354, 20305, 20294, 20291, 20286, 20284, 20233, 20223, 20213, 20213, 20189, 20180, 20166, 20136, 20119, 20118, 20077, 20075, 20051, 20043, 19937, 19907, 19884, 19628, 19596, 19565, 19457, 19440, 19439, 19393, 19336, 19286, 19273, 19234, 19223, 19199, 19172, 19146, 19141, 19134, 19041, 18983, 18973, 18963, 18930, 18903, 18842, 18800, 18736, 18703, 18644, 18639, 18635, 18585, 18568, 18507, 18492, 18488, 18420, 18394, 18330, 18245, 18164, 18152, 18061, 18009, 17942, 17936, 17836, 17776, 17771, 17752, 17744, 17721, 17686, 17668, 17658, 17601, 17568, 17543, 17540, 17484, 17468, 17390, 17348, 17337, 17323, 17297, 17297, 17261, 17254, 17201, 17111, 17044, 17042, 17040, 17033, 16927, 16918, 16899, 16899, 16797, 16768, 16749, 16700, 16698, 16688, 16681, 16668, 16652, 16622, 16616, 16579, 16539, 16532, 16504, 16477, 16462, 16458, 16420, 16336, 16285, 16244, 16233, 16096, 16094, 16033, 16002, 15951, 15933, 15901, 15888, 15829, 15829, 15821, 15768, 15746, 15730, 15695, 15613, 15599, 15578, 15466, 15426, 15375, 15370, 15355, 15318, 15276, 15249, 15229, 15228, 15190, 15185, 15153, 15150, 15102, 15077, 15073, 15062, 15031, 15008, 14975, 14943, 14933, 14812, 14811, 14804, 14769, 14753, 14738, 14707, 14689, 14674, 14662, 14644, 14592, 14573, 14543, 14540, 14519, 14504, 14377, 14367, 14357, 14326, 14296, 14251, 14188, 14173, 14150, 14133, 14084, 14046, 14005, 14005, 13973, 13964, 13957, 13925, 13914, 13903, 13854, 13715, 13698, 13696, 13628, 13581, 13565, 13477, 13463, 13462, 13449, 13371, 13234, 13092, 13012, 12955, 12947, 12931, 12896, 12878, 12849, 12830, 12825, 12703, 12701, 12678, 12639, 12410, 12389, 12347, 12305, 12268, 12172, 12106, 12066, 12062, 12060, 12038, 12023, 11946, 11925, 11922, 11913, 11866, 11806, 11804, 11777, 11701, 11700, 11609, 11531, 11523, 11513, 11445, 11427, 11404, 11400, 11388, 11379, 11374, 11348, 11340, 11299, 11292, 11273, 11254, 11168, 11095, 11035, 11025, 10974, 10971, 10909, 10904, 10894, 10842, 10716, 10670, 10633, 10627, 10581, 10573, 10552, 10510, 10478, 10465, 10446, 10415, 10408, 10333, 10149, 10143, 10135, 10066, 10055, 10021, 10020, 9989, 9984, 9984, 9951, 9946, 9908, 9901, 9889, 9868, 9859, 9830, 9784, 9777, 9753, 9725, 9722, 9692, 9659, 9601, 9559, 9512, 9424, 9401, 9272, 9251, 9238, 9217, 9182, 9178, 9173, 9169, 9080, 9024, 8994, 8990, 8976, 8893, 8885, 8869, 8797, 8778, 8734, 8622, 8496, 8403, 8367, 8355, 8353, 8352, 8330, 8327, 8313, 8298, 8273, 8273, 8249, 8237, 8187, 8180, 8173, 8146, 8125, 8089, 8087, 8051, 8029, 8015, 7904, 7871, 7859, 7840, 7820, 7819, 7641, 7595, 7581, 7574, 7572, 7561, 7534, 7527, 7497, 7466, 7416, 7406, 7386, 7385, 7309, 7277, 7275, 7273, 7230, 7211, 7164, 7162, 7123, 7121, 7108, 7107, 7042, 6986, 6913, 6817, 6807, 6798, 6775, 6773, 6732, 6691, 6622, 6545, 6527, 6519, 6450, 6445, 6433, 6337, 6328, 6326, 6323, 6313, 6256, 6223, 6222, 6194, 6194, 6173, 6173, 6157, 6148, 6074, 6032, 6028, 5969, 5965, 5938, 5936, 5908, 5877, 5804, 5803, 5767, 5719, 5596, 5490, 5484, 5443, 5438, 5416, 5410, 5365, 5346, 5345, 5310, 5244, 5194, 5142, 5121, 5015, 4999, 4970, 4961, 4910, 4901, 4873, 4851, 4774, 4749, 4707, 4695, 4654, 4624, 4617, 4614, 4607, 4590, 4563, 4522, 4513, 4498, 4470, 4435, 4379, 4343, 4338, 4309, 4271, 4244, 4199, 4195, 4157, 4156, 4156, 4154, 4142, 4132, 4103, 4037, 3998, 3886, 3872, 3820, 3783, 3767, 3691, 3678, 3673, 3663, 3662, 3583, 3580, 3558, 3540, 3485, 3463, 3401, 3389, 3358, 3356, 3353, 3302, 3287, 3285, 3260, 3244, 3241, 3167, 3162, 3157, 3014, 2947, 2877, 2817, 2786, 2764, 2674, 2622, 2608, 2561, 2482, 2450, 2449, 2444, 2401, 2385, 2383, 2366, 2357, 2325, 2321, 2304, 2293, 2251, 2240, 2198, 2177, 2160, 2135, 2112, 2088, 2010, 2009, 1945, 1907, 1887, 1885, 1838, 1829, 1822, 1724, 1712, 1708, 1663, 1651, 1648, 1608, 1598, 1534, 1445, 1389, 1369, 1319, 1281, 1273, 1254, 1246, 1216, 1196, 1195, 1181, 1124, 1092, 1092, 1091, 1089, 1052, 1026, 1002, 924, 897, 886, 886, 878, 857, 842, 830, 801, 799, 769, 756, 706, 626, 614, 525, 505, 458, 449, 431, 394, 383, 326, 325, 270, 165, 147, 142, 104, 97, 50, 41, -5, -5, -35, -41, -59, -74, -239, -338, -339, -345, -353, -425, -425, -459, -463, -465, -486, -566, -594, -598, -603, -638, -654, -685, -687, -706, -724, -751, -792, -801, -845, -864, -952, -993, -1023, -1028, -1053, -1058, -1086, -1120, -1132, -1146, -1152, -1172, -1189, -1243, -1277, -1289, -1295, -1297, -1324, -1359, -1385, -1392, -1397, -1408, -1416, -1418, -1428, -1444, -1445, -1468, -1492, -1503, -1507, -1538, -1542, -1560, -1560, -1698, -1721, -1734, -1758, -1758, -1848, -1849, -1880, -1884, -1890, -1942, -1976, -2020, -2059, -2091, -2114, -2122, -2140, -2188, -2202, -2253, -2294, -2316, -2332, -2367, -2389, -2493, -2560, -2577, -2594, -2604, -2637, -2649, -2704, -2713, -2763, -2768, -2805, -2844, -2845, -2849, -2899, -2907, -2921, -2963, -2968, -2988, -2991, -3056, -3063, -3129, -3144, -3158, -3258, -3269, -3271, -3343, -3343, -3366, -3384, -3406, -3424, -3437, -3477, -3507, -3525, -3532, -3592, -3661, -3672, -3719, -3774, -3774, -3846, -3860, -3891, -3915, -3918, -4031, -4076, -4077, -4108, -4169, -4211, -4319, -4374, -4382, -4483, -4596, -4704, -4789, -4790, -4818, -4846, -4848, -4960, -5000, -5105, -5124, -5142, -5170, -5221, -5308, -5345, -5365, -5455, -5606, -5610, -5630, -5632, -5656, -5671, -5676, -5747, -5877, -5884, -5898, -5904, -5979, -6016, -6091, -6096, -6161, -6242, -6258, -6303, -6317, -6335, -6364, -6367, -6534, -6562, -6597, -6660, -6687, -6695, -6739, -6760, -6826, -6906, -6921, -6928, -7015, -7019, -7061, -7123, -7132, -7142, -7149, -7162, -7163, -7188, -7192, -7233, -7286, -7353, -7388, -7397, -7411, -7435, -7479, -7502, -7564, -7574, -7577, -7592, -7646, -7679, -7687, -7704, -7732, -7752, -7870, -7876, -7926, -8003, -8013, -8077, -8078, -8110, -8153, -8166, -8278, -8306, -8474, -8490, -8505, -8516, -8594, -8614, -8683, -8717, -8746, -8749, -8756, -8824, -8863, -8890, -8945, -8954, -8969, -8985, -9054, -9059, -9159, -9164, -9199, -9230, -9256, -9449, -9512, -9547, -9577, -9599, -9641, -9659, -9697, -9785, -9815, -9833, -9863, -9865, -9943, -9947, -9967, -10012, -10062, -10069, -10079, -10093, -10108, -10296, -10302, -10347, -10355, -10357, -10366, -10394, -10410, -10467, -10480, -10532, -10553, -10598, -10620, -10646, -10653, -10654, -10719, -10747, -10759, -10761, -10762, -10817, -10913, -10919, -10930, -10950, -10951, -11048, -11109, -11149, -11169, -11172, -11222, -11297, -11300, -11314, -11419, -11446, -11448, -11472, -11495, -11552, -11613, -11634, -11671, -11729, -11760, -11767, -11789, -11790, -11825, -11867, -11868, -11885, -11933, -11948, -11964, -11965, -11975, -11975, -11978, -11992, -12004, -12025, -12051, -12056, -12119, -12151, -12159, -12178, -12184, -12201, -12232, -12237, -12316, -12363, -12373, -12406, -12420, -12445, -12460, -12477, -12589, -12598, -12609, -12621, -12671, -12691, -12738, -12790, -12791, -12807, -12835, -12839, -12944, -12960, -13032, -13048, -13089, -13098, -13212, -13221, -13224, -13233, -13253, -13296, -13298, -13298, -13376, -13384, -13393, -13395, -13520, -13521, -13572, -13590, -13600, -13621, -13677, -13729, -13746, -13800, -13827, -13849, -13859, -13939, -13962, -13964, -13966, -14008, -14057, -14069, -14125, -14135, -14138, -14144, -14147, -14189, -14223, -14260, -14268, -14286, -14289, -14301, -14325, -14389, -14407, -14408, -14450, -14456, -14524, -14564, -14626, -14715, -14792, -14813, -14820, -14847, -14850, -14867, -14881, -14912, -14919, -14924, -14945, -15029, -15039, -15112, -15135, -15150, -15175, -15201, -15203, -15250, -15257, -15327, -15348, -15350, -15372, -15395, -15404, -15465, -15535, -15577, -15578, -15589, -15670, -15757, -15828, -15832, -15966, -15981, -15994, -15995, -15999, -16008, -16055, -16064, -16077, -16138, -16243, -16283, -16303, -16310, -16350, -16354, -16388, -16508, -16530, -16534, -16561, -16569, -16584, -16599, -16627, -16701, -16730, -16760, -16764, -16773, -16778, -16792, -16819, -16843, -16894, -16917, -16930, -16960, -16973, -16985, -17044, -17104, -17120, -17124, -17142, -17147, -17154, -17159, -17195, -17233, -17241, -17250, -17302, -17337, -17371, -17419, -17422, -17431, -17475, -17477, -17525, -17547, -17677, -17725, -17732, -17819, -17819, -17825, -17830, -17929, -17935, -17936, -17948, -18076, -18139, -18247, -18261, -18263, -18280, -18287, -18290, -18348, -18359, -18377, -18458, -18479, -18480, -18536, -18537, -18561, -18574, -18619, -18679, -18690, -18692, -18694, -18792, -18792, -18797, -18806, -18808, -18814, -18837, -18865, -18873, -18889, -18915, -18926, -18929, -18959, -18967, -18978, -19001, -19021, -19025, -19093, -19098, -19123, -19134, -19146, -19148, -19158, -19179, -19197, -19211, -19233, -19237, -19266, -19409, -19440, -19463, -19528, -19528, -19548, -19568, -19587, -19656, -19691, -19702, -19756, -19767, -19767, -19769, -19852, -19880, -19908, -19925, -19963, -19982, -20062, -20087, -20117, -20133, -20159, -20163, -20211, -20212, -20250, -20256, -20314, -20332, -20335, -20361, -20368, -20438, -20453, -20496, -20510, -20542, -20559, -20694, -20710, -20714, -20725, -20806, -20883, -20920, -20972, -20990, -21010, -21014, -21017, -21039, -21061, -21096, -21104, -21134, -21226, -21297, -21338, -21388, -21428, -21465, -21556, -21606, -21622, -21669, -21722, -21726, -21772, -21780, -21790, -21816, -21821, -21846, -21857, -21955, -21994, -22071, -22121, -22144, -22148, -22156, -22215, -22248, -22321, -22372, -22374, -22386, -22452, -22490, -22573, -22612, -22623, -22624, -22642, -22709, -22732, -22760, -22766, -22774, -22808, -22876, -22930, -22934, -22954, -23030, -23057, -23076, -23109, -23136, -23310, -23320, -23337, -23363, -23366, -23383, -23399, -23570, -23573, -23579, -23662, -23668, -23670, -23689, -23810, -23856, -23906, -23953, -23959, -23966, -23975, -23987, -23995, -24053, -24063, -24083, -24090, -24096, -24162, -24194, -24206, -24273, -24327, -24395, -24403, -24436, -24463, -24489, -24549, -24582, -24628, -24654, -24664, -24674, -24677, -24695, -24783, -24798, -24838, -24841, -24881, -24906, -24924, -24926, -24926, -25026, -25028, -25087, -25130, -25172, -25173, -25200, -25211, -25246, -25256, -25276, -25285, -25334, -25361, -25364, -25373, -25428, -25469, -25477, -25567, -25606, -25610, -25619, -25634, -25641, -25821, -25886, -25918, -25930, -25943, -25948, -25956, -26108, -26118, -26172, -26175, -26190, -26207, -26232, -26245, -26254, -26332, -26370, -26377, -26413, -26415, -26459, -26490, -26540, -26619, -26720, -26743, -26763, -26792, -26812, -26848, -26874, -26890, -26938, -26945, -27035, -27057, -27057, -27149, -27165, -27181, -27232, -27271, -27313, -27387, -27404, -27442, -27460, -27463, -27463, -27473, -27490, -27494, -27561, -27567, -27592, -27602, -27737, -27785, -27850, -27852, -27860, -27874, -27880, -27900, -27913, -27919, -27926, -27946, -28000, -28018, -28059, -28068, -28088, -28092, -28094, -28120, -28131, -28162, -28163, -28202, -28285, -28305, -28308, -28330, -28355, -28362, -28368, -28373, -28398, -28400, -28419, -28463, -28490, -28504, -28555, -28566, -28594, -28594, -28623, -28680, -28758, -28775, -28797, -28834, -28853, -28884, -28941, -28950, -28976, -29025, -29039, -29046, -29097, -29118, -29141, -29179, -29181, -29184, -29196, -29197, -29210, -29231, -29242, -29317, -29339, -29342, -29351, -29356, -29406, -29423, -29425, -29431, -29479, -29482, -29495, -29537, -29581, -29619, -29622, -29627, -29633, -29660, -29710, -29715, -29755, -29765, -29774, -29782, -29786, -29814, -29851, -29883, -29951, -30085, -30098, -30131, -30222, -30232, -30256, -30275, -30283, -30306, -30319, -30383, -30395, -30399, -30436, -30437, -30469, -30494, -30535, -30536, -30564, -30572, -30665, -30711, -30742, -30768, -30809, -31015, -31031, -31057, -31077, -31177, -31260, -31270, -31280, -31296, -31372, -31416, -31552, -31589, -31635, -31656, -31699, -31713, -31727, -31780, -31919, -31979, -31981  .text  main:  jal sort  j end  sort:  la R16, array #This is a sudo instruction called "load address" that saves the initial address of array in register R16.  addi R8, R0, 0  addi R17, R0, 1999    sortLoop1:  bgt R8, R17, sortEndloop1  addi R9, R0, 0  sub R10, R17, R9    sortLoop2:  bgt R9, R10, sortEndloop2  sll R11, R9, 2  add R12, R16, R11  lw R13, 0(R12)  lw R14, 4(R12)  bgt R13, R14, swap  j endSwap    swap:  add R15, R13, R0  add R13, R14, R0  add R14, R15, R0  sw R13, 0(R12)  sw R14, 4(R12)    endSwap:  addi R9, R9, 1  j sortLoop2    sortEndloop2:  addi R8, R8, 1  j sortLoop1    sortEndloop1:  jr R31    end:  add R0, R0, R0 |