

Short ID	Problem 1 comments	Problem 1 points (75)	Problem 2 points (25)	Total
0336	Entirely correct	75	20	95
0365	Entirely correct	75	25	100
0408	Entirely correct.	75	25	100
0423	The code results in the original list of unsorted numbers. The code shows reasonable effort but does not work properly.	50	20	70
0522	Entirely correct	75	25	100
0596	Correctly sorts the first 8 numbers. A portion of the second half of the numbers are sorted correctly but are stored in the wrong indices.	60	25	85
0638	Entirely correct.	75	20	95
0822	Entirely correct	75	25	100
1086	Entirely correct	75	20	95
1247	Missing '\$' on lw instruction. After fixing, it worked correctly.	70	20	90
1329	Entirely correct	75	20	95
1971	Entirely correct.	75	20	95
2085	The code results in the original list of unsorted numbers. The code shows reasonable effort but does not work properly.	50	25	75
2227	Output is a list of 0's. Incorrectly call mergesort three times at each level. Never updates the \$a* registers for each recursive mergesort call. Merge function looks correct.	45	20	65
2291	Entirely correct	75	25	100
2483	Entirely correct	75	25	100
2736	Output is only slightly sorted. It also contains values which are not even from the input. Logic seems pretty good though. Couldn't get code to sort properly.	50	20	70
2756	Entirely correct.	75	25	100
2774	The program would not run because of subi instructions and incorrect syntax for lw instructions. After fixing these issues, the program still did not complete and exceptions such as "bad address in data/stack read" occurred.	30	20	50
3110	Program gave an "instruction references undefined symbol" error. Logic seems to be correct for the most part but program will not run. Could be trying to access indices which are out of range.	55	25	80
3184	Entirely correct.	75	20	95
3376	Entirely correct	75	25	100
3771	The code results in the original list of unsorted numbers. The code shows reasonable effort but does not work properly.	50	20	70
3804	The code shows good effort but would not execute. It threw an exception as a result of trying to load an invalid address into \$8.	50	25	75
4066	Entirely correct	75	25	100
4138	The code appears to be structured correctly but I could not get it to run. Upon attempting to execute, it threw an exception as a result of trying to load an invalid address	50	20	70
4398	Entirely correct	75	25	100
4437	The program does not finish executing. There is an error in the way he is handling the stack pointer.	55	25	80
4560	The program finishes execution but only a few values are stored in A and those that are, are not in the correct position. Other indices of A have a value of 0.	45	20	65
4616	Output is a list of 0's. Incorrectly call mergesort three times at each level. Never updates the \$a* registers for each recursive mergesort call. Merge function looks correct.	45	20	65
4709	The program enters an infinite loop and never completes.	45	25	70
5057	Entirely correct.	75	25	100
5479	The code results in the original list of unsorted numbers. The code shows reasonable effort but does not work properly.	50	25	75
5642	Entirely correct	75	20	95
5668	The program executes but after completion, none of the values are successfully written and the sorted list is not obtained. There may be an error in how the numbers are being stored.	45	20	65
5725	Correctly sorts the first 8 numbers. A portion of the second half of the numbers are sorted correctly but are stored in the wrong indices.	60	25	85
5768	Entirely correct.	75	25	100
5849	Lab 2 directory exists but lab2.s file is empty	0	0	0
5866	The program executes but the second half of the array remains unsorted and the first half contains miscellaneous values.	55	25	80
5901	Program would not run. Syntax was an issue but his logic seemed to be there for the most part.	45	20	65
6416	Entirely correct	75	25	100
6550	The code results in the original list of unsorted numbers. The code shows reasonable effort but does not work properly.	50	10	60
7770	Entirely correct	75	25	100
7808	Entirely correct.	75	20	95
8022	The code shows good effort but would not execute. It threw an exception as a result of trying to load an invalid address into \$8.	50	25	75
8037	Almost entirely correct. Not completely sorted. As Justin mentioned, I think the issue is with handling of \$sp.	70	15	85
8178	Code is not recursive, doesn't call mergesort from main. Only 60 lines of code included (which is hard to follow).	20	10	30
8554	Entirely correct	75	20	95
8861	Entirely correct.	75	25	100
8939	Entirely correct.	75	25	100
9131	Entirely correct	75	25	100
9229	The program executes but after completion, none of the values are successfully written and the sorted list is not obtained. There may be an error in how the numbers are being stored.	45	20	65
9322	Output is only slightly sorted. It also contains values which are not even from the input. Logic seems pretty good though. Couldn't get code to sort properly.	50	20	70
9353	Almost entirely correct. Not completely sorted.	70	20	90
9640	The program finishes execution but only a few values are stored in A and those that are, are not in the correct position. Other indices of A have a value of 0.	45	20	65
9691	Entirely correct.	75	25	100
		61.25	21.70	82.95