

Project Debriefing

- What was the easiest part of the project?
- What was the hardest part?
- Of which part of your program are you proudest?
- Which part would you re-write first,
if you had the chance to improve *only one* part?
- What would you do differently on your *next* project?
- What could I do differently to improve the course?

Your answers can refer to programming practices, language tools, project management, teamwork, content,

Historical Comparison

2013-nco	Java	much	2019-arg	Java	some
2013-rack	Racket	all+	2019-cob	Python	all
2013-sgt	Java	all	2019-fork	Python	all
			2019-rtt	Python	most
2015-cdb	Python	some	2019-rack	Racket	all*
2015-sm	Racket	some			
2015-em	Python	all+	2021-3_t	Python	some
			2021-opt	Python	some-
2016-com	Java	some-	2021-stgs	Python	all*
2016-asw	C++	some-	2021-snks	Python	all*
2016-af	Rust	none	2021-rack	Racket	none
2017-def	C++	all	2022-tma	Java	all*
2017-lex	Python	all	2022-jt	Java	some*
2017-duk	Java	some			
			2025-cs	Python	all
2018-ava	Python	some-	2025-com	Java	all*
2018-mww	Python	most	2025-kl	Python	most
2018-unc	Python	some	2025-lc	Python	all
2018-wcf	Python	all+	2025-wr	Zig	all

Size of Compiler — *source code*

			lines	words	bytes
2013-rack	Racket	all+	4650	17450	185523
2015-em	Python	all+	6682	14497	189581
2019-arg	Java	some	4591	17953	187080
2019-cob	Python	all+	2709	8366	98823
2019-fork	Python	all	1945	6578	78905
2019-rtt	Python	most	2221	8502	101693
2019-rack	Racket	all*	951	4953	40526*
2021-3_t	Python	some	1711	4821	61970
2021-opt	Python	some-	1519	4302	49748
2021-stgs	Python	all*	3132	9553	132982
2021-snks	Python	all*	4557	10980	152087
2025-cs	Python	all	4299	10925	143451
2025-com	Java	all*	4773	14414	162368
2025-kl	Python	most	1587	6704	74110
2025-lc	Python	all	1995	5664	69988
2025-wr	Zig	all	2538	10713	122657

Using the Compiler

euclid 100 15

	time to compile	size of tm	stmts exec	run time
2013-rack	1.196s	151	724	0.003s
2015-em	0.454s	152	630	0.002s
2025-cs	0.538s	116	527	0.028s
2025-com	0.627s	112	539	0.542s
2025-kl	0.223s			
2025-lc	0.307s	132	556	0.612s
2025-wr	0.014s	130	573	0.378s

Using the Compiler

sieve 40

	time to compile	size of tm	stmts exec	run time	
2013-rack	1.198s	371	208046	0.005s	(40)
2015-em	0.084s	324	41164	0.003s	(30)
2025-cs	0.071s	274	51297	0.713s	(20)
2025-com	0.104s	251	52000	0.031s	(40)
2025-kl	0.056s	228	125679	0.546s	err
2025-lc	0.052s	281	54592	0.532s	(40)
2025-wr	0.010s	293	100956	0.354s	(31)

Using the Compiler

is-excellent 3468

	time to compile	size of tm	stmts exec	run time
--	--------------------	---------------	---------------	-------------

2013-rack

2015-em

2025-cs	0.073s	389	831	0.028s
2025-com	0.101s	355	797	0.028s
2025-kl	0.054s			
2025-lc	0.050s	394	842	0.029s
2025-wr	0.010s	436	856	0.027s

Challenge: Short-Circuit and

	exe size -----	output -----	short circuit? -----	
2013-rack				
2015-em				
2025-cs	74	0 0	yes	
2025-com	54	0 0	yes	
2025-kl	50	0 1 0	no	
2025-lc	73	0 0	yes	
2025-wr	61	1 0 0	no	*err

Challenge: Short-Circuit or

	exe size -----	output -----	short circuit? -----	
2013-rack				
2015-em				
2025-cs	74	1 1	yes	
2025-com	54	1 1	yes	
2025-kl	50	1 0 1	no	
2025-lc	73	1 1	yes	
2025-wr	61	0 1 1	no	*err

Challenge: factorial

factorial 25 => 2076180480

	exe size -----	stmts exec -----	run- time -----
2013-rack	106	1491	0.044s
2015-em	117	1274	0.039s
2025-cs	87	981	0.030s
2025-com	68	972	0.029s
2025-kl	66	890	0.028s
2025-lc	86	1028	0.028s
2025-wr	73	995	0.029s

Challenge: Stack Size

tm stack-size

	exe size -----	stack depth -----	stmts exec -----
2013-rack	40	inf	inf
2015-em	66	126	4714
2025-cs	40	84	1447
2025-com	28	169	2949
2025-kl	38	254	5876
2025-lc	--		
2025-wr	33	113	1818

Challenge: Stack Size

tm-ext stack-size

	exe size -----	stack depth -----	stmts exec -----
2013-rack	40	inf	inf
2015-em	66	131070	4849642
2025-cs	40	87380	1485479
2025-com	28	174761	2621429
2025-kl	38	262142	6029300
2025-lc	--		
2025-wr	33	116508	1864135

A New Program

tm-ext carmichael n

	time to compile	size of tm	stmts exec	run time	n
2025-cs	0.077s	367	120305527	1.128s	8911
2025-com	0.104s	355	1834	0.041s	err
2025-kl	0.055s				
2025-lc	0.051s	398	130371345	1.089s	8911
2025-wr	0.011s	418	830287244	6.764s	8911

Team Debriefing

- How well did each person contribute to the overall success of the project?
- For a rating, give a score between 1-10 as a summary evaluation of the person's contribution. *No two team members may receive the same rating.*
- For points, divide the 100% available to your project among your team members according to what you think each person earned. *Team members **may** receive the same numbers of points.*

Answers can refer to any aspect of the project. Be specific.

<The End>

•