TRACE32 Export

1. Table of source files

address	source file
ST:00000400	\\sieve\crt0\crt0.sx
SR:00000460	\\sieve\isr\isr.c
ST:000004B0	\\sieve\sieve\sieve.c

2. BookMarks

bookmark	address	symbol	source file	line	comment / justification
1	C:000013D2	\\sieve\sieve\main\47	\\sieve\".\src\sieve.c"\sieve.c	733	j is never 0, seems to be useless code
nop1	C:0000077A				padding NOP
bad1	C:00001638	\\sieve\sieve\background	\\sieve\".\src\sieve.c"\sieve.c	819	never called => dead code
bad2	C:00000ED8	\\sieve\sieve\func27	\\sieve\".\src\sieve.c"\sieve.c	555	never called => dead code

3. COVerage.ListModule

address	tree	coverage	executed	0% 50% 100%	branches	ok	taken	not taken	never	bytes	bytesok
P:0400--045F	\\sieve\crt0	never	0.000%		0.000%	0	0	0	4	96	0
P:0460--04AF	\\sieve\isr	never	0.000%		-	0	0	0	-	80	0
P:04B0--167B	\\sieve\sieve	partial	75.109%		78.125%	47	4	2	11	4556	3422
P:0400--167B	total	partial	72.316%		73.529%	47	4	2	15	4732	3422

4. COVerage.ListFunc

address	tree	coverage	executed	0%50%100%	branches	ok	taken	not taken	never	bytes	bytesok
P:0460--04AF	\\sieve\isr	never	0.000%		-	0	0	0	-	80	0
P:0460--0487	\\sieve\isr\exc_isr	never	0.000%		-	0	0	0	-	40	0
P:0488--04AF	\\sieve\isr\exc_snoop	never	0.000%		-	0	0	0	-	40	0
P:04B0--167B	\\sieve\sieve	partial	75.109%		78.125%	47	4	2	11	4556	3422
P:04B0--04BB	\\sieve\sieve\func0	never	0.000%		-	0	0	0	-	12	0
P:04BC--04D7	\\sieve\sieve\func1	ok	100.000%		-	0	0	0	-	28	28
P:04D8--0563	\\sieve\sieve\func2	partial	88.571%		75.000%	1	1	0	0	140	124
P:0564--05BF	\\sieve\sieve\func2a	partial	97.826%		100.000%	1	0	0	0	92	90
P:05C0--05FF	\\sieve\sieve\func2b	partial	96.875%		100.000%	1	0	0	0	64	62
P:0600--06BF	\\sieve\sieve\func2c	partial	96.875%		100.000%	2	0	0	0	192	186
P:06C0--071B	\\sieve\sieve\func2d	partial	97.826%		100.000%	1	0	0	0	92	90
P:071C--072B	\\sieve\sieve\func3	ok	100.000%		-	0	0	0	-	16	16
P:072C--0779	\\sieve\sieve\func4	ok	100.000%		-	0	0	0	-	78	78
P:077C--07A7	\\sieve\sieve\func5	ok	100.000%		-	0	0	0	-	44	44
P:07A8--07F3	\\sieve\sieve\func6	partial	100.000%		-	0	0	0	-	76	76
P:07F4--086B	\\sieve\sieve\func7	partial	100.000%		-	0	0	0	-	120	120
P:086C--0A2B	\\sieve\sieve\func8	partial	99.553%		-	0	0	0	-	448	446
P:0A2C--0A87	\\sieve\sieve\func9	partial	100.000%		100.000%	2	0	0	0	92	92
P:0A88--0C27	\\sieve\sieve\func10	ok	100.000%		100.000%	33	0	0	0	416	416
P:0C28--0C93	\\sieve\sieve\func11	partial	40.740%		50.000%	0	0	1	0	108	44
P:0C94--0CAB	\\sieve\sieve\func12	never	0.000%		-	0	0	0	-	24	0
P:0CAC--0D09	\\sieve\sieve\func13	ok	100.000%		100.000%	1	0	0	0	94	94
P:0D0C--0D29	\\sieve\sieve\func14	ok	100.000%		-	0	0	0	-	30	30
P:0D2C--0D4B	\\sieve\sieve\func15	ok	100.000%		-	0	0	0	-	32	32
P:0D4C--0D63	\\sieve\sieve\func16	ok	100.000%		-	0	0	0	-	24	24
P:0D64--0D8F	\\sieve\sieve\func17	ok	100.000%		-	0	0	0	-	44	44
P:0D90--0DB3	\\sieve\sieve\func18	ok	100.000%		-	0	0	0	-	36	36
P:0DB4--0DD7	\\sieve\sieve\func19	ok	100.000%		-	0	0	0	-	36	36
P:0DD8--0E0B	\\sieve\sieve\func20	ok	100.000%		-	0	0	0	-	52	52
P:0E0C--0E39	\\sieve\sieve\func21	ok	100.000%		-	0	0	0	-	46	46
P:0E3C--0E6B	\\sieve\sieve\func22	ok	100.000%		-	0	0	0	-	48	48
P:0E6C--0E9D	\\sieve\sieve\func23	never	0.000%		-	0	0	0	-	50	0
P:0EA0--0EAF	\\sieve\sieve\func24	never	0.000%		-	0	0	0	-	16	0
P:0EB0--0EC3	\\sieve\sieve\func25	never	0.000%		-	0	0	0	-	20	0
P:0EC4--0ED7	\\sieve\sieve\func26	never	0.000%		-	0	0	0	-	20	0
P:0ED8--0F17	\\sieve\sieve\func27	never	0.000%		-	0	0	0	-	64	0
P:0F18--0F57	\\sieve\sieve\test_cond_instr	never	0.000%		0.000%	0	0	0	1	64	0
P:0F58--103B	\\sieve\sieve\func_sin	partial	97.368%		100.000%	2	0	0	0	228	222
P:103C--111F	\\sieve\sieve\init_linked_list	partial	100.000%		100.000%	3	0	0	0	228	228
P:1120--1167	\\sieve\sieve\subst	never	0.000%		0.000%	0	0	0	1	72	0
P:1168--120B	\\sieve\sieve\encode	never	0.000%		0.000%	0	0	0	3	164	0
P:120C--126B	\\sieve\sieve\func28	never	0.000%		-	0	0	0	-	96	0
P:126C--15DF	\\sieve\sieve\main	partial	69.909%		40.000%	0	3	1	1	884	618
P:15E0--1637	\\sieve\sieve\sieve	never	0.000%		0.000%	0	0	0	4	88	0
P:1638--167B	\\sieve\sieve\background	never	0.000%		0.000%	0	0	0	1	68	0
P:0460--167B	total	partial	73.813%		78.125%	47	4	2	11	4636	3422

5. COVerage.ListLine

address	tree	coverage	executed	0%50%100%	branches	ok	taken	not taken	never	bytes	bytesok
P:0460--04AF	\\sieve\isr	never	0.000%		-	0	0	0	-	80	0
P:0460--0487	\\sieve\isr\exc_isr	never	0.000%		-	0	0	0	-	40	0
P:0460--046B	isr.c \1--10	never	0.000%		-	0	0	0	-	12	0
P:046C--046F	isr.c \11--11	never	0.000%		-	0	0	0	-	4	0
P:0470--047B	isr.c \12--12	never	0.000%		-	0	0	0	-	12	0
P:047C--0487	isr.c \13--13	never	0.000%		-	0	0	0	-	12	0
P:0488--04AF	\\sieve\isr\exc_snoop	never	0.000%		-	0	0	0	-	40	0
P:0488--048F	isr.c \14--22	never	0.000%		-	0	0	0	-	8	0
P:0490--0493	isr.c \23--23	never	0.000%		-	0	0	0	-	4	0
P:0494--049F	isr.c \24--24	never	0.000%		-	0	0	0	-	12	0
P:04A0--04AF	isr.c \25--25	never	0.000%		-	0	0	0	-	16	0
P:04B0--167B	\\sieve\sieve	partial	75.109%		78.125%	47	4	2	11	4556	3422
P:04B0--04BB	\\sieve\sieve\func0	never	0.000%		-	0	0	0	-	12	0
P:04B0--04B3	sieve.c \1--144	never	0.000%		-	0	0	0	-	4	0
P:04B4--04BB	sieve.c \145--145	never	0.000%		-	0	0	0	-	8	0
P:04BC--04D7	\\sieve\sieve\func1	ok	100.000%		-	0	0	0	-	28	28
P:04BC--04C3	sieve.c \146--148	ok	100.000%		-	0	0	0	-	8	8
P:04C4--04CD	sieve.c \149--149	ok	100.000%		-	0	0	0	-	10	10
P:04CE--04D7	sieve.c \150--150	ok	100.000%		-	0	0	0	-	10	10
P:04D8--0563	\\sieve\sieve\func2	partial	88.571%		75.000%	1	1	0	0	140	124
P:04D8--04DD	sieve.c \151--153	ok	100.000%		-	0	0	0	-	6	6
P:04DE--04E3	sieve.c \154--159	ok	100.000%		-	0	0	0	-	6	6
P:04E4--04E9	sieve.c \160--160	ok	100.000%		-	0	0	0	-	6	6
P:04EA--04F1	sieve.c \161--162	ok	100.000%		-	0	0	0	-	8	8
P:04F2--04F9	sieve.c \163--163	ok	100.000%		-	0	0	0	-	8	8
P:04FA--04FD	sieve.c \164--165	ok	100.000%		-	0	0	0	-	4	4
P:04FE--050D	sieve.c \166--166	ok	100.000%		-	0	0	0	-	16	16
P:050E--0513	sieve.c \164--165	ok	100.000%		100.000%	1	0	0	0	6	6
P:0514--0521	sieve.c \167--168	ok	100.000%		-	0	0	0	-	14	14
P:0522--052B	sieve.c \169--169	ok	100.000%		-	0	0	0	-	10	10
P:052C--0535	sieve.c \170--171	taken	80.000%		50.000%	0	1	0	0	10	8
P:0536--053D	sieve.c \173--174	never	0.000%		-	0	0	0	-	8	0
P:053E--0541	sieve.c \175--176	never	0.000%		-	0	0	0	-	4	0
P:0542--0543	sieve.c \172--172	ok	100.000%		-	0	0	0	-	2	2
P:0544--0563	sieve.c \177--177	partial	93.750%		-	0	0	0	-	32	30
P:0564--05BF	\\sieve\sieve\func2a	partial	97.826%		100.000%	1	0	0	0	92	90
P:0564--0569	sieve.c \178--180	ok	100.000%		-	0	0	0	-	6	6
P:056A--0577	sieve.c \181--184	ok	100.000%		-	0	0	0	-	14	14
P:0578--0581	sieve.c \185--185	ok	100.000%		-	0	0	0	-	10	10
P:0582--0585	sieve.c \186--187	ok	100.000%		-	0	0	0	-	4	4
P:0586--05A1	sieve.c \188--188	ok	100.000%		-	0	0	0	-	28	28
P:05A2--05AB	sieve.c \186--187	ok	100.000%		100.000%	1	0	0	0	10	10
P:05AC--05BF	sieve.c \189--189	partial	90.000%		-	0	0	0	-	20	18
P:05C0--05FF	\\sieve\sieve\func2b	partial	96.875%		100.000%	1	0	0	0	64	62
P:05C0--05C5	sieve.c \190--192	ok	100.000%		-	0	0	0	-	6	6
P:05C6--05CB	sieve.c \193--196	ok	100.000%		-	0	0	0	-	6	6
P:05CC--05D1	sieve.c \197--197	ok	100.000%		-	0	0	0	-	6	6
P:05D2--05D5	sieve.c \198--199	ok	100.000%		-	0	0	0	-	4	4
P:05D6--05E5	sieve.c \200--200	ok	100.000%		-	0	0	0	-	16	16
P:05E6--05EB	sieve.c \198--199	ok	100.000%		100.000%	1	0	0	0	6	6
P:05EC--05FF	sieve.c \201--201	partial	90.000%		-	0	0	0	-	20	18
P:0600--06BF	\\sieve\sieve\func2c	partial	96.875%		100.000%	2	0	0	0	192	186
P:0600--0605	sieve.c \202--204	ok	100.000%		-	0	0	0	-	6	6
P:0606--061B	sieve.c \205--208	ok	100.000%		-	0	0	0	-	22	22
P:061C--062F	sieve.c \209--209	ok	100.000%		-	0	0	0	-	20	20
P:0630--0635	sieve.c \210--211	ok	100.000%		-	0	0	0	-	6	6
P:0636--065D	sieve.c \212--212	ok	100.000%		-	0	0	0	-	40	40
P:065E--068F	sieve.c \210--211	ok	100.000%		100.000%	2	0	0	0	50	50
P:0690--06BF	sieve.c \213--213	partial	87.500%		-	0	0	0	-	48	42
P:06C0--071B	\\sieve\sieve\func2d	partial	97.826%		100.000%	1	0	0	0	92	90
P:06C0--06C5	sieve.c \214--216	ok	100.000%		-	0	0	0	-	6	6
P:06C6--06D3	sieve.c \217--220	ok	100.000%		-	0	0	0	-	14	14
P:06D4--06DD	sieve.c \221--221	ok	100.000%		-	0	0	0	-	10	10
P:06DE--06E1	sieve.c \222--223	ok	100.000%		-	0	0	0	-	4	4

address	tree	coverage	executed	0%50%100%	branches	ok	taken	not taken	never	bytes	bytesok
P:06F2--06F7	sieve.c \224--224	ok	100.000%		100.000%	0	0	0	0	22	22
P:06F8--0707	sieve.c \222--223	ok	100.000%		100.000%	1	0	0	0	16	16
P:0708--071B	sieve.c \225--225	partial	90.000%		-	0	0	0	-	20	18
P:071C--072B	\\sieve\sieve\func3	ok	100.000%		-	0	0	0	-	16	16
P:071C--071F	sieve.c \226--229	ok	100.000%		-	0	0	0	-	4	4
P:0720--0721	sieve.c \230--230	ok	100.000%		-	0	0	0	-	2	2
P:0722--072B	sieve.c \231--231	ok	100.000%		-	0	0	0	-	10	10
P:072C--0779	\\sieve\sieve\func4	ok	100.000%		-	0	0	0	-	78	78
P:072C--0743	sieve.c \232--234	ok	100.000%		-	0	0	0	-	24	24
P:0744--0759	sieve.c \235--235	ok	100.000%		-	0	0	0	-	22	22
P:075A--076B	sieve.c \236--237	ok	100.000%		-	0	0	0	-	18	18
P:076C--077B	sieve.c \238--238	partial	87.500%		-	0	0	0	-	16	14
P:077C--07A7	\\sieve\sieve\func5	ok	100.000%		-	0	0	0	-	44	44
P:077C--078D	sieve.c \239--241	ok	100.000%		-	0	0	0	-	18	18
P:078E--079B	sieve.c \242--242	ok	100.000%		-	0	0	0	-	14	14
P:079C--07A7	sieve.c \243--243	ok	100.000%		-	0	0	0	-	12	12
P:07A8--07F3	\\sieve\sieve\func6	partial	100.000%		-	0	0	0	-	76	76
P:07A8--07B1	sieve.c \244--246	ok	100.000%		-	0	0	0	-	10	10
P:07B2--07B7	sieve.c \247--247	ok	100.000%		-	0	0	0	-	6	6
P:07B8--07BD	sieve.c \248--248	ok	100.000%		-	0	0	0	-	6	6
P:07BE--07C3	sieve.c \249--249	ok	100.000%		-	0	0	0	-	6	6
P:07C4--07C9	sieve.c \250--250	ok	100.000%		-	0	0	0	-	6	6
P:07CA--07D3	sieve.c \251--252	ok	100.000%		-	0	0	0	-	10	10
P:07D4--07F3	sieve.c \253--253	partial	100.000%		-	0	0	0	-	32	32
P:07F4--086B	\\sieve\sieve\func7	partial	100.000%		-	0	0	0	-	120	120
P:07F4--0801	sieve.c \254--256	ok	100.000%		-	0	0	0	-	14	14
P:0802--080B	sieve.c \257--257	ok	100.000%		-	0	0	0	-	10	10
P:080C--0815	sieve.c \258--258	ok	100.000%		-	0	0	0	-	10	10
P:0816--081F	sieve.c \259--259	ok	100.000%		-	0	0	0	-	10	10
P:0820--0829	sieve.c \260--260	ok	100.000%		-	0	0	0	-	10	10
P:082A--0839	sieve.c \261--262	ok	100.000%		-	0	0	0	-	16	16
P:083A--086B	sieve.c \263--263	partial	100.000%		-	0	0	0	-	50	50
P:086C--0A2B	\\sieve\sieve\func8	partial	99.553%		-	0	0	0	-	448	446
P:086C--086F	sieve.c \264--290	ok	100.000%		-	0	0	0	-	4	4
P:0870--0879	sieve.c \291--291	ok	100.000%		-	0	0	0	-	10	10
P:087A--0887	sieve.c \292--292	ok	100.000%		-	0	0	0	-	14	14
P:0888--0895	sieve.c \293--293	ok	100.000%		-	0	0	0	-	14	14
P:0896--08A3	sieve.c \294--294	ok	100.000%		-	0	0	0	-	14	14
P:08A4--08B3	sieve.c \295--295	ok	100.000%		-	0	0	0	-	16	16
P:08B4--08C1	sieve.c \296--296	ok	100.000%		-	0	0	0	-	14	14
P:08C2--08CD	sieve.c \297--297	ok	100.000%		-	0	0	0	-	12	12
P:08CE--08DB	sieve.c \298--298	ok	100.000%		-	0	0	0	-	14	14
P:08DC--08E7	sieve.c \299--299	ok	100.000%		-	0	0	0	-	12	12
P:08E8--08F5	sieve.c \300--300	ok	100.000%		-	0	0	0	-	14	14
P:08F6--08FF	sieve.c \301--301	ok	100.000%		-	0	0	0	-	10	10
P:0900--090D	sieve.c \302--302	ok	100.000%		-	0	0	0	-	14	14
P:090E--091B	sieve.c \303--303	ok	100.000%		-	0	0	0	-	14	14
P:091C--0921	sieve.c \304--304	ok	100.000%		-	0	0	0	-	6	6
P:0922--0927	sieve.c \305--305	ok	100.000%		-	0	0	0	-	6	6
P:0928--0937	sieve.c \306--306	ok	100.000%		-	0	0	0	-	16	16
P:0938--0945	sieve.c \307--307	ok	100.000%		-	0	0	0	-	14	14
P:0946--094F	sieve.c \308--308	ok	100.000%		-	0	0	0	-	10	10
P:0950--0959	sieve.c \309--310	ok	100.000%		-	0	0	0	-	10	10
P:095A--0963	sieve.c \311--311	ok	100.000%		-	0	0	0	-	10	10
P:0964--096D	sieve.c \312--312	ok	100.000%		-	0	0	0	-	10	10
P:096E--0979	sieve.c \313--313	ok	100.000%		-	0	0	0	-	12	12
P:097A--0985	sieve.c \314--314	ok	100.000%		-	0	0	0	-	12	12
P:0986--098F	sieve.c \315--315	ok	100.000%		-	0	0	0	-	10	10
P:0990--099B	sieve.c \316--316	ok	100.000%		-	0	0	0	-	12	12
P:099C--09A5	sieve.c \317--317	ok	100.000%		-	0	0	0	-	10	10
P:09A6--09B1	sieve.c \318--318	ok	100.000%		-	0	0	0	-	12	12
P:09B2--09BD	sieve.c \319--319	ok	100.000%		-	0	0	0	-	12	12
P:09BE--09C7	sieve.c \320--320	ok	100.000%		-	0	0	0	-	10	10
P:09C8--09D1	sieve.c \321--321	ok	100.000%		-	0	0	0	-	10	10
P:09D2--09DD	sieve.c \322--322	ok	100.000%		-	0	0	0	-	12	12
P:09DE--09E5	sieve.c \323--323	ok	100.000%		-	0	0	0	-	8	8
P:09E6--09EB	sieve.c \324--324	ok	100.000%		-	0	0	0	-	6	6

address	tree	coverage	executed	0%50%100%	branches	ok	taken	not taken	never	bytes	bytesok
P:09EC--09F5	sieve.c \325--325	ok	100.000%		-	0	0	0	-	10	10
P:09F6--09FF	sieve.c \326--326	ok	100.000%		-	0	0	0	-	10	10
P:0A00--0A09	sieve.c \327--327	ok	100.000%		-	0	0	0	-	10	10
P:0A0A--0A2B	sieve.c \328--328	partial	94.117%		-	0	0	0	-	34	32
P:0A2C--0A87	\\sieve\sieve\func9	partial	100.000%		100.000%	2	0	0	0	92	92
P:0A2C--0A31	sieve.c \329--331	ok	100.000%		-	0	0	0	-	6	6
P:0A32--0A37	sieve.c \332--336	ok	100.000%		-	0	0	0	-	6	6
P:0A38--0A3B	sieve.c \337--338	ok	100.000%		-	0	0	0	-	4	4
P:0A3C--0A41	sieve.c \339--344	ok	100.000%		-	0	0	0	-	6	6
P:0A42--0A45	sieve.c \345--346	ok	100.000%		-	0	0	0	-	4	4
P:0A46--0A4D	sieve.c \347--348	ok	100.000%		-	0	0	0	-	8	8
P:0A4E--0A55	sieve.c \349--349	ok	100.000%		-	0	0	0	-	8	8
P:0A56--0A5D	sieve.c \350--350	ok	100.000%		-	0	0	0	-	8	8
P:0A5E--0A65	sieve.c \351--351	ok	100.000%		-	0	0	0	-	8	8
P:0A66--0A6B	sieve.c \345--346	ok	100.000%		100.000%	1	0	0	0	6	6
P:0A6C--0A71	sieve.c \337--338	ok	100.000%		100.000%	1	0	0	0	6	6
P:0A72--0A73	sieve.c \352--355	ok	100.000%		-	0	0	0	-	2	2
P:0A74--0A87	sieve.c \356--356	partial	100.000%		-	0	0	0	-	20	20
P:0A88--0C27	\\sieve\sieve\func10	ok	100.000%		100.000%	33	0	0	0	416	416
P:0A88--0A8B	sieve.c \357--359	ok	100.000%		-	0	0	0	-	4	4
P:0A8C--0A8D	sieve.c \360--364	ok	100.000%		-	0	0	0	-	2	2
P:0A8E--0A91	sieve.c \365--365	ok	100.000%		-	0	0	0	-	4	4
P:0A92--0A93	sieve.c \366--366	ok	100.000%		-	0	0	0	-	2	2
P:0A94--0A99	sieve.c \365--365	ok	100.000%		100.000%	1	0	0	0	6	6
P:0A9A--0A9D	sieve.c \367--368	ok	100.000%		-	0	0	0	-	4	4
P:0A9E--0A9F	sieve.c \369--369	ok	100.000%		-	0	0	0	-	2	2
P:0AA0--0AA5	sieve.c \367--368	ok	100.000%		100.000%	1	0	0	0	6	6
P:0AA6--0AA9	sieve.c \370--370	ok	100.000%		-	0	0	0	-	4	4
P:0AAA--0AAB	sieve.c \371--371	ok	100.000%		-	0	0	0	-	2	2
P:0AAC--0AB1	sieve.c \370--370	ok	100.000%		100.000%	1	0	0	0	6	6
P:0AB2--0AB5	sieve.c \372--372	ok	100.000%		-	0	0	0	-	4	4
P:0AB6--0AB7	sieve.c \373--373	ok	100.000%		-	0	0	0	-	2	2
P:0AB8--0ABD	sieve.c \372--372	ok	100.000%		100.000%	1	0	0	0	6	6
P:0ABE--0AC1	sieve.c \374--374	ok	100.000%		-	0	0	0	-	4	4
P:0AC2--0AC3	sieve.c \375--375	ok	100.000%		-	0	0	0	-	2	2
P:0AC4--0AC9	sieve.c \374--374	ok	100.000%		100.000%	1	0	0	0	6	6
P:0ACA--0ACD	sieve.c \376--376	ok	100.000%		-	0	0	0	-	4	4
P:0ACE--0ACF	sieve.c \377--377	ok	100.000%		-	0	0	0	-	2	2
P:0AD0--0AD5	sieve.c \376--376	ok	100.000%		100.000%	1	0	0	0	6	6
P:0AD6--0AD9	sieve.c \378--378	ok	100.000%		-	0	0	0	-	4	4
P:0ADA--0ADB	sieve.c \379--379	ok	100.000%		-	0	0	0	-	2	2
P:0ADC--0AE1	sieve.c \378--378	ok	100.000%		100.000%	1	0	0	0	6	6
P:0AE2--0AE5	sieve.c \380--380	ok	100.000%		-	0	0	0	-	4	4
P:0AE6--0AE7	sieve.c \381--381	ok	100.000%		-	0	0	0	-	2	2
P:0AE8--0AED	sieve.c \380--380	ok	100.000%		100.000%	1	0	0	0	6	6
P:0AEE--0AF1	sieve.c \382--382	ok	100.000%		-	0	0	0	-	4	4
P:0AF2--0AF3	sieve.c \383--383	ok	100.000%		-	0	0	0	-	2	2
P:0AF4--0AF9	sieve.c \382--382	ok	100.000%		100.000%	1	0	0	0	6	6
P:0AFA--0AFD	sieve.c \384--384	ok	100.000%		-	0	0	0	-	4	4
P:0AFE--0AFF	sieve.c \385--385	ok	100.000%		-	0	0	0	-	2	2
P:0B00--0B05	sieve.c \384--384	ok	100.000%		100.000%	1	0	0	0	6	6
P:0B06--0B09	sieve.c \386--386	ok	100.000%		-	0	0	0	-	4	4
P:0B0A--0B0B	sieve.c \387--387	ok	100.000%		-	0	0	0	-	2	2
P:0B0C--0B11	sieve.c \386--386	ok	100.000%		100.000%	1	0	0	0	6	6
P:0B12--0B15	sieve.c \388--388	ok	100.000%		-	0	0	0	-	4	4
P:0B16--0B17	sieve.c \389--389	ok	100.000%		-	0	0	0	-	2	2
P:0B18--0B1D	sieve.c \388--388	ok	100.000%		100.000%	1	0	0	0	6	6
P:0B1E--0B21	sieve.c \390--390	ok	100.000%		-	0	0	0	-	4	4
P:0B22--0B23	sieve.c \391--391	ok	100.000%		-	0	0	0	-	2	2
P:0B24--0B29	sieve.c \390--390	ok	100.000%		100.000%	1	0	0	0	6	6
P:0B2A--0B2D	sieve.c \392--392	ok	100.000%		-	0	0	0	-	4	4
P:0B2E--0B2F	sieve.c \393--393	ok	100.000%		-	0	0	0	-	2	2
P:0B30--0B35	sieve.c \392--392	ok	100.000%		100.000%	1	0	0	0	6	6
P:0B36--0B39	sieve.c \394--394	ok	100.000%		-	0	0	0	-	4	4
P:0B3A--0B3B	sieve.c \395--395	ok	100.000%		-	0	0	0	-	2	2
P:0B3C--0B41	sieve.c \394--394	ok	100.000%		100.000%	1	0	0	0	6	6
P:0B42--0B45	sieve.c \396--396	ok	100.000%		-	0	0	0	-	4	4
P:0B46--0B47	sieve.c \397--397		100.000%		-	0	0	0	-	2	2

address	tree	coverage	executed	0%50%100%	branches	ok 1	taken 0	not taken 0	never 0	bytes 6	bytes ok 6
P:0B48--0B4D	sieve.c \396--396	ok	100.000%		100.000%	1	0	0	0	6	6
P:0B4E--0B51	sieve.c \398--398	ok	100.000%		-	0	0	0	-	4	4
P:0B52--0B53	sieve.c \399--399	ok	100.000%		-	0	0	0	-	2	2
P:0B54--0B59	sieve.c \398--398	ok	100.000%		100.000%	1	0	0	0	6	6
P:0B5A--0B5B	sieve.c \400--401	ok	100.000%		-	0	0	0	-	2	2
P:0B5C--0B5F	sieve.c \402--403	ok	100.000%		-	0	0	0	-	4	4
P:0B60--0B61	sieve.c \404--404	ok	100.000%		-	0	0	0	-	2	2
P:0B62--0B67	sieve.c \402--403	ok	100.000%		100.000%	1	0	0	0	6	6
P:0B68--0B6B	sieve.c \405--405	ok	100.000%		-	0	0	0	-	4	4
P:0B6C--0B6D	sieve.c \406--406	ok	100.000%		-	0	0	0	-	2	2
P:0B6E--0B73	sieve.c \405--405	ok	100.000%		100.000%	1	0	0	0	6	6
P:0B74--0B77	sieve.c \407--407	ok	100.000%		-	0	0	0	-	4	4
P:0B78--0B79	sieve.c \408--408	ok	100.000%		-	0	0	0	-	2	2
P:0B7A--0B7F	sieve.c \407--407	ok	100.000%		100.000%	1	0	0	0	6	6
P:0B80--0B83	sieve.c \409--409	ok	100.000%		-	0	0	0	-	4	4
P:0B84--0B85	sieve.c \410--410	ok	100.000%		-	0	0	0	-	2	2
P:0B86--0B8B	sieve.c \409--409	ok	100.000%		100.000%	1	0	0	0	6	6
P:0B8C--0B8F	sieve.c \411--411	ok	100.000%		-	0	0	0	-	4	4
P:0B90--0B91	sieve.c \412--412	ok	100.000%		-	0	0	0	-	2	2
P:0B92--0B97	sieve.c \411--411	ok	100.000%		100.000%	1	0	0	0	6	6
P:0B98--0B9B	sieve.c \413--413	ok	100.000%		-	0	0	0	-	4	4
P:0B9C--0B9D	sieve.c \414--414	ok	100.000%		-	0	0	0	-	2	2
P:0B9E--0BA3	sieve.c \413--413	ok	100.000%		100.000%	1	0	0	0	6	6
P:0BA4--0BA7	sieve.c \415--415	ok	100.000%		-	0	0	0	-	4	4
P:0BA8--0BA9	sieve.c \416--416	ok	100.000%		-	0	0	0	-	2	2
P:0BAA--0BAF	sieve.c \415--415	ok	100.000%		100.000%	1	0	0	0	6	6
P:0BB0--0BB3	sieve.c \417--417	ok	100.000%		-	0	0	0	-	4	4
P:0BB4--0BB5	sieve.c \418--418	ok	100.000%		-	0	0	0	-	2	2
P:0BB6--0BBB	sieve.c \417--417	ok	100.000%		100.000%	1	0	0	0	6	6
P:0BBC--0BBF	sieve.c \419--419	ok	100.000%		-	0	0	0	-	4	4
P:0BC0--0BC1	sieve.c \420--420	ok	100.000%		-	0	0	0	-	2	2
P:0BC2--0BC7	sieve.c \419--419	ok	100.000%		100.000%	1	0	0	0	6	6
P:0BC8--0BCB	sieve.c \421--421	ok	100.000%		-	0	0	0	-	4	4
P:0BCC--0BCD	sieve.c \422--422	ok	100.000%		-	0	0	0	-	2	2
P:0BCE--0BD3	sieve.c \421--421	ok	100.000%		100.000%	1	0	0	0	6	6
P:0BD4--0BD7	sieve.c \423--423	ok	100.000%		-	0	0	0	-	4	4
P:0BD8--0BD9	sieve.c \424--424	ok	100.000%		-	0	0	0	-	2	2
P:0BDA--0BDF	sieve.c \423--423	ok	100.000%		100.000%	1	0	0	0	6	6
P:0BE0--0BE3	sieve.c \425--425	ok	100.000%		-	0	0	0	-	4	4
P:0BE4--0BE5	sieve.c \426--426	ok	100.000%		-	0	0	0	-	2	2
P:0BE6--0BEB	sieve.c \425--425	ok	100.000%		100.000%	1	0	0	0	6	6
P:0BEC--0BEF	sieve.c \427--427	ok	100.000%		-	0	0	0	-	4	4
P:0BF0--0BF1	sieve.c \428--428	ok	100.000%		-	0	0	0	-	2	2
P:0BF2--0BF7	sieve.c \427--427	ok	100.000%		100.000%	1	0	0	0	6	6
P:0BF8--0BFB	sieve.c \429--429	ok	100.000%		-	0	0	0	-	4	4
P:0BFC--0BFD	sieve.c \430--430	ok	100.000%		-	0	0	0	-	2	2
P:0BFE--0C03	sieve.c \429--429	ok	100.000%		100.000%	1	0	0	0	6	6
P:0C04--0C07	sieve.c \431--431	ok	100.000%		-	0	0	0	-	4	4
P:0C08--0C09	sieve.c \432--432	ok	100.000%		-	0	0	0	-	2	2
P:0C0A--0C0F	sieve.c \431--431	ok	100.000%		100.000%	1	0	0	0	6	6
P:0C10--0C13	sieve.c \433--433	ok	100.000%		-	0	0	0	-	4	4
P:0C14--0C15	sieve.c \434--434	ok	100.000%		-	0	0	0	-	2	2
P:0C16--0C1B	sieve.c \433--433	ok	100.000%		100.000%	1	0	0	0	6	6
P:0C1C--0C1D	sieve.c \435--436	ok	100.000%		-	0	0	0	-	2	2
P:0C1E--0C27	sieve.c \437--437	ok	100.000%		-	0	0	0	-	10	10
P:0C28--0C93	\\sieve\sieve\func11	partial	40.740%		50.000%	0	0	1	0	108	44
P:0C28--0C2F	sieve.c \438--440	ok	100.000%		-	0	0	0	-	8	8
P:0C30--0C41	sieve.c \441--441	not taken	88.888%		50.000%	0	0	1	0	18	16
P:0C42--0C47	sieve.c \442--444	never	0.000%		-	0	0	0	-	6	0
P:0C48--0C4D	sieve.c \445--445	never	0.000%		-	0	0	0	-	6	0
P:0C4E--0C55	sieve.c \446--446	never	0.000%		-	0	0	0	-	8	0
P:0C56--0C5D	sieve.c \447--448	never	0.000%		-	0	0	0	-	8	0
P:0C5E--0C61	sieve.c \449--450	never	0.000%		-	0	0	0	-	4	0
P:0C62--0C67	sieve.c \451--452	never	0.000%		-	0	0	0	-	6	0
P:0C68--0C6D	sieve.c \453--453	never	0.000%		-	0	0	0	-	6	0
P:0C6E--0C75	sieve.c \454--454	never	0.000%		-	0	0	0	-	8	0
P:0C76--0C7D	sieve.c \455--458	never	0.000%		-	0	0	0	-	8	0

address	tree	coverage	executed	0%50%100%	branches	ok	taken	not taken	never	bytes	bytesok
P:0C7E--0C7F	sieve.c \459--460	ok	100.000%		-	0	0	0	-	2	2
P:0C80--0C81	sieve.c \461--462	ok	100.000%		-	0	0	0	-	2	2
P:0C82--0C93	sieve.c \463--463	partial	88.888%		-	0	0	0	-	18	16
P:0C94--0CAB	\\sieve\sieve\func12	never	0.000%		-	0	0	0	-	24	0
P:0C94--0C9B	sieve.c \464--469	never	0.000%		-	0	0	0	-	8	0
P:0C9C--0CA1	sieve.c \470--470	never	0.000%		-	0	0	0	-	6	0
P:0CA2--0CAB	sieve.c \471--471	never	0.000%		-	0	0	0	-	10	0
P:0CAC--0D09	\\sieve\sieve\func13	ok	100.000%		100.000%	1	0	0	0	94	94
P:0CAC--0CB7	sieve.c \472--474	ok	100.000%		-	0	0	0	-	12	12
P:0CB8--0CC3	sieve.c \475--477	ok	100.000%		-	0	0	0	-	12	12
P:0CC4--0CCB	sieve.c \478--478	ok	100.000%		-	0	0	0	-	8	8
P:0CCC--0CD3	sieve.c \479--479	ok	100.000%		-	0	0	0	-	8	8
P:0CD4--0CD9	sieve.c \480--481	ok	100.000%		100.000%	1	0	0	0	6	6
P:0CDA--0CF3	sieve.c \482--482	ok	100.000%		-	0	0	0	-	26	26
P:0CF4--0CFD	sieve.c \483--484	ok	100.000%		-	0	0	0	-	10	10
P:0CFE--0D0B	sieve.c \485--485	partial	85.714%		-	0	0	0	-	14	12
P:0D0C--0D29	\\sieve\sieve\func14	ok	100.000%		-	0	0	0	-	30	30
P:0D0C--0D17	sieve.c \486--488	ok	100.000%		-	0	0	0	-	12	12
P:0D18--0D1D	sieve.c \489--489	ok	100.000%		-	0	0	0	-	6	6
P:0D1E--0D2B	sieve.c \490--490	partial	85.714%		-	0	0	0	-	14	12
P:0D2C--0D4B	\\sieve\sieve\func15	ok	100.000%		-	0	0	0	-	32	32
P:0D2C--0D37	sieve.c \491--493	ok	100.000%		-	0	0	0	-	12	12
P:0D38--0D3F	sieve.c \494--494	ok	100.000%		-	0	0	0	-	8	8
P:0D40--0D4B	sieve.c \495--495	ok	100.000%		-	0	0	0	-	12	12
P:0D4C--0D63	\\sieve\sieve\func16	ok	100.000%		-	0	0	0	-	24	24
P:0D4C--0D53	sieve.c \496--498	ok	100.000%		-	0	0	0	-	8	8
P:0D54--0D57	sieve.c \499--499	ok	100.000%		-	0	0	0	-	4	4
P:0D58--0D63	sieve.c \500--500	ok	100.000%		-	0	0	0	-	12	12
P:0D64--0D8F	\\sieve\sieve\func17	ok	100.000%		-	0	0	0	-	44	44
P:0D64--0D75	sieve.c \501--503	ok	100.000%		-	0	0	0	-	18	18
P:0D76--0D83	sieve.c \504--504	ok	100.000%		-	0	0	0	-	14	14
P:0D84--0D8F	sieve.c \505--505	ok	100.000%		-	0	0	0	-	12	12
P:0D90--0DB3	\\sieve\sieve\func18	ok	100.000%		-	0	0	0	-	36	36
P:0D90--0D9D	sieve.c \506--508	ok	100.000%		-	0	0	0	-	14	14
P:0D9E--0DA7	sieve.c \509--509	ok	100.000%		-	0	0	0	-	10	10
P:0DA8--0DB3	sieve.c \510--510	ok	100.000%		-	0	0	0	-	12	12
P:0DB4--0DD7	\\sieve\sieve\func19	ok	100.000%		-	0	0	0	-	36	36
P:0DB4--0DC1	sieve.c \511--513	ok	100.000%		-	0	0	0	-	14	14
P:0DC2--0DCB	sieve.c \514--514	ok	100.000%		-	0	0	0	-	10	10
P:0DCC--0DD7	sieve.c \515--515	ok	100.000%		-	0	0	0	-	12	12
P:0DD8--0E0B	\\sieve\sieve\func20	ok	100.000%		-	0	0	0	-	52	52
P:0DD8--0DE9	sieve.c \516--518	ok	100.000%		-	0	0	0	-	18	18
P:0DEA--0DFF	sieve.c \519--519	ok	100.000%		-	0	0	0	-	22	22
P:0E00--0E0B	sieve.c \520--520	ok	100.000%		-	0	0	0	-	12	12
P:0E0C--0E39	\\sieve\sieve\func21	ok	100.000%		-	0	0	0	-	46	46
P:0E0C--0E1B	sieve.c \521--523	ok	100.000%		-	0	0	0	-	16	16
P:0E1C--0E2D	sieve.c \524--524	ok	100.000%		-	0	0	0	-	18	18
P:0E2E--0E3B	sieve.c \525--525	partial	85.714%		-	0	0	0	-	14	12
P:0E3C--0E6B	\\sieve\sieve\func22	ok	100.000%		-	0	0	0	-	48	48
P:0E3C--0E4D	sieve.c \526--528	ok	100.000%		-	0	0	0	-	18	18
P:0E4E--0E5F	sieve.c \529--529	ok	100.000%		-	0	0	0	-	18	18
P:0E60--0E6B	sieve.c \530--530	ok	100.000%		-	0	0	0	-	12	12
P:0E6C--0E9D	\\sieve\sieve\func23	never	0.000%		-	0	0	0	-	50	0
P:0E6C--0E7F	sieve.c \531--533	never	0.000%		-	0	0	0	-	20	0
P:0E80--0E91	sieve.c \534--534	never	0.000%		-	0	0	0	-	18	0
P:0E92--0E9F	sieve.c \535--535	never	0.000%		-	0	0	0	-	14	0
P:0EA0--0EAF	\\sieve\sieve\func24	never	0.000%		-	0	0	0	-	16	0
P:0EA0--0EA3	sieve.c \536--538	never	0.000%		-	0	0	0	-	4	0
P:0EA4--0EA5	sieve.c \539--539	never	0.000%		-	0	0	0	-	2	0
P:0EA6--0EAF	sieve.c \540--540	never	0.000%		-	0	0	0	-	10	0
P:0EB0--0EC3	\\sieve\sieve\func25	never	0.000%		-	0	0	0	-	20	0
P:0EB0--0EB3	sieve.c \541--543	never	0.000%		-	0	0	0	-	4	0
P:0EB4--0EB5	sieve.c \544--544	never	0.000%		-	0	0	0	-	2	0
P:0EB6--0EC3	sieve.c \545--545	never	0.000%		-	0	0	0	-	14	0
P:0EC4--0ED7	\\sieve\sieve\func26	never	0.000%		-	0	0	0	-	20	0
P:0EC4--0EC7	sieve.c \546--548	never	0.000%		-	0	0	0	-	4	0
P:0EC8--0EC9	sieve.c \549--551	never	0.000%		-	0	0	0	-	2	0

address	trace	coverage	executed	0%50%100%	branches	ok	taken	not taken	never	bytes	bytesok
P:0ECA--0ED7	sieve.c \552--552	never	0.000%		-	0	0	0	-	14	0
P:0ED8--0EE7	sieve.c \553--555	never	0.000%		-	0	0	0	-	16	0
P:0EE8--0EF9	sieve.c \556--556	never	0.000%		-	0	0	0	-	18	0
P:0EFA--0F05	sieve.c \557--558	never	0.000%		-	0	0	0	-	12	0
P:0F06--0F17	sieve.c \559--559	never	0.000%		-	0	0	0	-	18	0
P:0F18--0F57	\\sieve\sieve\test_cond_instr	never	0.000%		0.000%	0	0	0	1	64	0
P:0F18--0F21	sieve.c \560--567	never	0.000%		-	0	0	0	-	10	0
P:0F22--0F29	sieve.c \568--580	never	0.000%		-	0	0	0	-	8	0
P:0F2A--0F2F	sieve.c \581--581	never	0.000%		0.000%	0	0	0	1	6	0
P:0F30--0F37	sieve.c \582--582	never	0.000%		-	0	0	0	-	8	0
P:0F38--0F3D	sieve.c \583--583	never	0.000%		-	0	0	0	-	6	0
P:0F3E--0F45	sieve.c \584--584	never	0.000%		-	0	0	0	-	8	0
P:0F46--0F49	sieve.c \585--586	never	0.000%		-	0	0	0	-	4	0
P:0F4A--0F4B	sieve.c \587--589	never	0.000%		-	0	0	0	-	2	0
P:0F4C--0F57	sieve.c \590--590	never	0.000%		-	0	0	0	-	12	0
P:0F58--103B	\\sieve\sieve\func_sin	partial	97.368%		100.000%	2	0	0	0	228	222
P:0F58--0F5D	sieve.c \591--596	ok	100.000%		-	0	0	0	-	6	6
P:0F5E--0F61	sieve.c \597--600	ok	100.000%		-	0	0	0	-	4	4
P:0F62--0F6B	sieve.c \601--601	ok	100.000%		-	0	0	0	-	10	10
P:0F6C--0FCD	sieve.c \602--602	ok	100.000%		-	0	0	0	-	98	98
P:0FCE--0FFF	sieve.c \601--601	ok	100.000%		100.000%	2	0	0	0	50	50
P:1000--103B	sieve.c \603--603	partial	90.000%		-	0	0	0	-	60	54
P:103C--111F	\\sieve\sieve\init_linked_list	partial	100.000%		100.000%	3	0	0	0	228	228
P:103C--1041	sieve.c \604--606	ok	100.000%		-	0	0	0	-	6	6
P:1042--1045	sieve.c \607--607	ok	100.000%		-	0	0	0	-	4	4
P:1046--1049	sieve.c \608--608	ok	100.000%		-	0	0	0	-	4	4
P:104A--104F	sieve.c \609--611	ok	100.000%		-	0	0	0	-	6	6
P:1050--1055	sieve.c \612--612	ok	100.000%		-	0	0	0	-	6	6
P:1056--105B	sieve.c \613--613	ok	100.000%		-	0	0	0	-	6	6
P:105C--1061	sieve.c \614--614	ok	100.000%		-	0	0	0	-	6	6
P:1062--1067	sieve.c \615--616	ok	100.000%		-	0	0	0	-	6	6
P:1068--1097	sieve.c \617--617	ok	100.000%		100.000%	1	0	0	0	48	48
P:1098--10C5	sieve.c \618--618	ok	100.000%		100.000%	1	0	0	0	46	46
P:10C6--10D9	sieve.c \619--619	ok	100.000%		-	0	0	0	-	20	20
P:10DA--10FB	sieve.c \620--620	ok	100.000%		-	0	0	0	-	34	34
P:10FC--1109	sieve.c \615--616	ok	100.000%		100.000%	1	0	0	0	14	14
P:110A--111F	sieve.c \621--622	partial	100.000%		-	0	0	0	-	22	22
P:1120--1167	\\sieve\sieve\subst	never	0.000%		0.000%	0	0	0	1	72	0
P:1120--112B	sieve.c \623--625	never	0.000%		-	0	0	0	-	12	0
P:112C--113F	sieve.c \626--626	never	0.000%		0.000%	0	0	0	1	20	0
P:1140--1143	sieve.c \627--628	never	0.000%		-	0	0	0	-	4	0
P:1144--1147	sieve.c \629--630	never	0.000%		-	0	0	0	-	4	0
P:1148--114B	sieve.c \631--632	never	0.000%		-	0	0	0	-	4	0
P:114C--114F	sieve.c \633--634	never	0.000%		-	0	0	0	-	4	0
P:1150--1153	sieve.c \635--636	never	0.000%		-	0	0	0	-	4	0
P:1154--1157	sieve.c \637--638	never	0.000%		-	0	0	0	-	4	0
P:1158--1167	sieve.c \639--639	never	0.000%		-	0	0	0	-	16	0
P:1168--120B	\\sieve\sieve\encode	never	0.000%		0.000%	0	0	0	3	164	0
P:1168--116F	sieve.c \640--642	never	0.000%		-	0	0	0	-	8	0
P:1170--1175	sieve.c \643--645	never	0.000%		-	0	0	0	-	6	0
P:1176--1183	sieve.c \646--646	never	0.000%		-	0	0	0	-	14	0
P:1184--1197	sieve.c \647--647	never	0.000%		0.000%	0	0	0	2	20	0
P:1198--11A5	sieve.c \648--648	never	0.000%		-	0	0	0	-	14	0
P:11A6--11B9	sieve.c \649--649	never	0.000%		-	0	0	0	-	20	0
P:11BA--11C9	sieve.c \650--650	never	0.000%		-	0	0	0	-	16	0
P:11CA--11DD	sieve.c \651--652	never	0.000%		-	0	0	0	-	20	0
P:11DE--11EB	sieve.c \653--654	never	0.000%		-	0	0	0	-	14	0
P:11EC--11FD	sieve.c \643--645	never	0.000%		0.000%	0	0	0	1	18	0
P:11FE--11FF	sieve.c \655--656	never	0.000%		-	0	0	0	-	2	0
P:1200--120B	sieve.c \657--657	never	0.000%		-	0	0	0	-	12	0
P:120C--126B	\\sieve\sieve\func28	never	0.000%		-	0	0	0	-	96	0
P:120C--1215	sieve.c \658--669	never	0.000%		-	0	0	0	-	10	0
P:1216--1223	sieve.c \670--670	never	0.000%		-	0	0	0	-	14	0
P:1224--1239	sieve.c \671--671	never	0.000%		-	0	0	0	-	22	0
P:123A--125F	sieve.c \672--672	never	0.000%		-	0	0	0	-	38	0
P:1260--126B	sieve.c \673--673	never	0.000%		-	0	0	0	-	12	0
P:126C--15DF	\\sieve\sieve\main	partial	69.909%		40.000%	0	3	1	1	884	618

address	tree	coverage	executed	0%50%100%	branches	ok	taken	not taken	never	bytes	bytesok
P:126C--1271	sieve.c \674--686	ok	100.000%		-	0	0	0	-	6	6
P:1272--1275	sieve.c \687--691	ok	100.000%		-	0	0	0	-	4	4
P:1276--127D	sieve.c \692--694	taken	75.000%		50.000%	0	1	0	0	8	6
P:127E--1285	sieve.c \695--695	never	0.000%		-	0	0	0	-	8	0
P:1286--128B	sieve.c \696--697	ok	100.000%		-	0	0	0	-	6	6
P:128C--1291	sieve.c \698--698	ok	100.000%		-	0	0	0	-	6	6
P:1292--129B	sieve.c \699--699	ok	100.000%		-	0	0	0	-	10	10
P:129C--12B1	sieve.c \700--701	ok	100.000%		-	0	0	0	-	22	22
P:12B2--12E1	sieve.c \702--702	partial	79.166%		50.000%	0	2	0	0	48	38
P:12E2--1309	sieve.c \703--703	ok	100.000%		-	0	0	0	-	40	40
P:130A--131F	sieve.c \704--704	ok	100.000%		-	0	0	0	-	22	22
P:1320--1325	sieve.c \705--706	ok	100.000%		-	0	0	0	-	6	6
P:1326--132B	sieve.c \707--707	ok	100.000%		-	0	0	0	-	6	6
P:132C--1331	sieve.c \708--708	ok	100.000%		-	0	0	0	-	6	6
P:1332--1337	sieve.c \709--709	ok	100.000%		-	0	0	0	-	6	6
P:1338--133B	sieve.c \710--711	ok	100.000%		-	0	0	0	-	4	4
P:133C--133F	sieve.c \712--712	ok	100.000%		-	0	0	0	-	4	4
P:1340--1343	sieve.c \713--713	ok	100.000%		-	0	0	0	-	4	4
P:1344--1347	sieve.c \714--714	ok	100.000%		-	0	0	0	-	4	4
P:1348--134B	sieve.c \715--715	ok	100.000%		-	0	0	0	-	4	4
P:134C--1351	sieve.c \716--717	ok	100.000%		-	0	0	0	-	6	6
P:1352--1357	sieve.c \718--718	ok	100.000%		-	0	0	0	-	6	6
P:1358--135B	sieve.c \719--720	ok	100.000%		-	0	0	0	-	4	4
P:135C--1361	sieve.c \721--721	ok	100.000%		-	0	0	0	-	6	6
P:1362--138F	sieve.c \722--722	ok	100.000%		-	0	0	0	-	46	46
P:1390--139D	sieve.c \723--723	ok	100.000%		-	0	0	0	-	14	14
P:139E--13C3	sieve.c \724--725	ok	100.000%		-	0	0	0	-	38	38
P:13C4--13D1	sieve.c \726--727	ok	100.000%		-	0	0	0	-	14	14
P:13D2--13D3	sieve.c \733--733	never	0.000%		-	0	0	0	-	2	0
P:13D4--13E3	sieve.c \728--730	ok	100.000%		-	0	0	0	-	16	16
P:13E4--13E9	sieve.c \731--732	not taken	66.666%		50.000%	0	0	1	0	6	4
P:13EA--13EF	sieve.c \734--735	ok	100.000%		-	0	0	0	-	6	6
P:13F0--13F5	sieve.c \736--737	ok	100.000%		-	0	0	0	-	6	6
P:13F6--1403	sieve.c \738--738	ok	100.000%		-	0	0	0	-	14	14
P:1404--140D	sieve.c \739--739	ok	100.000%		-	0	0	0	-	10	10
P:140E--1423	sieve.c \740--740	ok	100.000%		-	0	0	0	-	22	22
P:1424--1427	sieve.c \741--742	ok	100.000%		-	0	0	0	-	4	4
P:1428--142B	sieve.c \743--743	ok	100.000%		-	0	0	0	-	4	4
P:142C--142F	sieve.c \744--744	ok	100.000%		-	0	0	0	-	4	4
P:1430--1435	sieve.c \745--745	ok	100.000%		-	0	0	0	-	6	6
P:1436--143F	sieve.c \746--746	ok	100.000%		-	0	0	0	-	10	10
P:1440--1445	sieve.c \747--747	ok	100.000%		-	0	0	0	-	6	6
P:1446--144B	sieve.c \748--748	ok	100.000%		-	0	0	0	-	6	6
P:144C--1451	sieve.c \749--749	ok	100.000%		-	0	0	0	-	6	6
P:1452--1459	sieve.c \750--750	ok	100.000%		-	0	0	0	-	8	8
P:145A--1461	sieve.c \751--751	ok	100.000%		-	0	0	0	-	8	8
P:1462--1469	sieve.c \752--752	ok	100.000%		-	0	0	0	-	8	8
P:146A--1473	sieve.c \753--753	ok	100.000%		-	0	0	0	-	10	10
P:1474--147D	sieve.c \754--754	ok	100.000%		-	0	0	0	-	10	10
P:147E--1487	sieve.c \755--755	ok	100.000%		-	0	0	0	-	10	10
P:1488--1491	sieve.c \756--756	ok	100.000%		-	0	0	0	-	10	10
P:1492--1499	sieve.c \757--758	never	0.000%		-	0	0	0	-	8	0
P:149A--14A3	sieve.c \759--759	never	0.000%		-	0	0	0	-	10	0
P:14A4--14AB	sieve.c \760--760	never	0.000%		-	0	0	0	-	8	0
P:14AC--14C9	sieve.c \761--761	never	0.000%		-	0	0	0	-	30	0
P:14CA--14D1	sieve.c \762--763	never	0.000%		-	0	0	0	-	8	0
P:14D2--14D5	sieve.c \764--764	never	0.000%		-	0	0	0	-	4	0
P:14D6--14E3	sieve.c \765--766	never	0.000%		-	0	0	0	-	14	0
P:14E4--14F1	sieve.c \767--767	never	0.000%		-	0	0	0	-	14	0
P:14F2--14F7	sieve.c \768--769	never	0.000%		-	0	0	0	-	6	0
P:14F8--14FD	sieve.c \770--770	never	0.000%		-	0	0	0	-	6	0
P:14FE--1503	sieve.c \771--771	never	0.000%		-	0	0	0	-	6	0
P:1504--150B	sieve.c \772--772	never	0.000%		-	0	0	0	-	8	0
P:150C--1511	sieve.c \773--773	never	0.000%		-	0	0	0	-	6	0
P:1512--1517	sieve.c \774--774	never	0.000%		-	0	0	0	-	6	0
P:1518--151D	sieve.c \775--775	never	0.000%		-	0	0	0	-	6	0
P:151E--1523	sieve.c \776--776	never	0.000%		-	0	0	0	-	6	0

address	tree	coverage	executed	0%50%100%	branches	ok	taken	not taken	never	bytes	bytes ok
P:1524--1529	sieve.c \777--777	never	0.000%		-	0	0	0	-	6	0
P:152A--152F	sieve.c \778--778	never	0.000%		-	0	0	0	-	6	0
P:1530--1535	sieve.c \779--779	never	0.000%		-	0	0	0	-	6	0
P:1536--153B	sieve.c \780--780	never	0.000%		-	0	0	0	-	6	0
P:153C--1541	sieve.c \781--781	never	0.000%		-	0	0	0	-	6	0
P:1542--1547	sieve.c \782--782	never	0.000%		-	0	0	0	-	6	0
P:1548--1551	sieve.c \783--784	never	0.000%		0.000%	0	0	0	1	10	0
P:1552--1553	sieve.c \785--786	never	0.000%		-	0	0	0	-	2	0
P:1554--15DF	sieve.c \787--787	p-read	65.714%		-	0	0	0	-	140	92
P:15E0--1637	\\sieve\sieve\sieve	never	0.000%		0.000%	0	0	0	4	88	0
P:15E0--15E5	sieve.c \788--794	never	0.000%		-	0	0	0	-	6	0
P:15E6--15E9	sieve.c \795--798	never	0.000%		-	0	0	0	-	4	0
P:15EA--15F9	sieve.c \799--800	never	0.000%		0.000%	0	0	0	1	16	0
P:15FA--15FD	sieve.c \801--802	never	0.000%		-	0	0	0	-	4	0
P:15FE--1605	sieve.c \803--803	never	0.000%		0.000%	0	0	0	1	8	0
P:1606--1609	sieve.c \804--804	never	0.000%		-	0	0	0	-	4	0
P:160A--160B	sieve.c \805--805	never	0.000%		-	0	0	0	-	2	0
P:160C--160D	sieve.c \806--806	never	0.000%		-	0	0	0	-	2	0
P:160E--1613	sieve.c \807--807	never	0.000%		-	0	0	0	-	6	0
P:1614--1615	sieve.c \808--808	never	0.000%		-	0	0	0	-	2	0
P:1616--1619	sieve.c \806--806	never	0.000%		0.000%	0	0	0	1	4	0
P:161A--161F	sieve.c \809--810	never	0.000%		-	0	0	0	-	6	0
P:1620--1625	sieve.c \801--802	never	0.000%		0.000%	0	0	0	1	6	0
P:1626--1627	sieve.c \811--814	never	0.000%		-	0	0	0	-	2	0
P:1628--1637	sieve.c \815--815	never	0.000%		-	0	0	0	-	16	0
P:1638--167B	\\sieve\sieve\background	never	0.000%		0.000%	0	0	0	1	68	0
P:1638--163B	sieve.c \816--819	never	0.000%		-	0	0	0	-	4	0
P:163C--1649	sieve.c \820--822	never	0.000%		-	0	0	0	-	14	0
P:164A--164F	sieve.c \823--825	never	0.000%		-	0	0	0	-	6	0
P:1650--1651	sieve.c \826--826	never	0.000%		-	0	0	0	-	2	0
P:1652--165B	sieve.c \827--827	never	0.000%		-	0	0	0	-	10	0
P:165C--1663	sieve.c \826--826	never	0.000%		0.000%	0	0	0	1	8	0
P:1664--166D	sieve.c \828--828	never	0.000%		-	0	0	0	-	10	0
P:166E--167D	sieve.c \829--829	partial	12.500%		-	0	0	0	-	16	2
P:0460--167B	**total**	partial	73.813%		78.125%	47	4	2	11	4636	3422

6. COVerage.ListVar

address	tree	coverage	read	0% 100%	write	0% 100%	bytes	read	write
P:0x4B0	\\sieve\sieve								
D:4428--4443	init_linked_list\text	p-read	0.000%		0.000%		28	0	0
D:4628--462B	period	read	0.000%		0.000%		4	0	0
D:4650--465A	func26\x1	never	0.000%		0.000%		11	0	0
D:465C--465F	func2\fstatic	readwrite	0.000%		100.000%		4	0	4
D:46C0--46C3	mstatic1	readwrite	0.000%		100.000%		4	0	4
D:46C4--46C7	mstatic2	readwrite	0.000%		100.000%		4	0	4
D:46C8--46CB	mcount	readwrite	0.000%		100.000%		4	0	4
D:46D0--46D3	background\bcnt2	never	0.000%		0.000%		4	0	0
D:46D4--46D7	background\bcnt1	never	0.000%		0.000%		4	0	0
D:46D8--46DB	func9\stat1	readwrite	0.000%		100.000%		4	0	4
D:46DC--46DF	func9\stat2	readwrite	0.000%		100.000%		4	0	4
D:46E0--46E3	func2\fstatic2	write	0.000%		100.000%		4	0	4
P:0x1680	\\sieve\monitor								
D:4660--4663	Monitor_Handler\Monitor_AddressLow	never	0.000%		0.000%		4	0	0
D:4664--46A3	Monitor_Handler\Monitor_Buffer	never	0.000%		0.000%		64	0	0
D:46A4--46A7	Monitor_Handler\Monitor_Index	never	0.000%		0.000%		4	0	0
D:46A8--46AB	Monitor_Handler\Monitor_Count	never	0.000%		0.000%		4	0	0
D:46AC--46AF	Monitor_Handler\Monitor_AddressHigh	never	0.000%		0.000%		4	0	0
P:0x208C	\\sieve\itm								
D:46B0--46B3	ItmMessage\init	never	0.000%		0.000%		4	0	0
	\\sieve\Global								
D:4360--4370	cstr1	never	0.000%		0.000%		17	0	0
D:4444--4447	ITM_BASE	never	0.000%		0.000%		4	0	0
D:4620--4624	defaultstring	never	0.000%		0.000%		5	0	0
D:462C--462D	plot1	readwrite	0.000%		100.000%		2	0	2
D:4630--4633	vfloat	readwrite	0.000%		100.000%		4	0	4
D:4638--463F	vdouble	readwrite	0.000%		100.000%		8	0	8
D:4640--4643	vint	never	0.000%		0.000%		4	0	0
D:4644--4648	vdiarray	never	0.000%		0.000%		5	0	0
D:464C--464F	monHook	read	0.000%		0.000%		4	0	0
D:46CC--46CD	plot2	write	0.000%		100.000%		2	0	2
D:46E8--4705	vdblarray	never	0.000%		0.000%		30	0	0
D:4708--470B	vpppulong	never	0.000%		0.000%		4	0	0
D:470C--470F	vpushort	never	0.000%		0.000%		4	0	0
D:4710--4713	vppppulong	never	0.000%		0.000%		4	0	0
D:4714--4717	vpchar	never	0.000%		0.000%		4	0	0
D:4718--471F	def	never	0.000%		0.000%		8	0	0
D:4720--4733	vtdef2	never	0.000%		0.000%		20	0	0
D:4734--473B	vunion	never	0.000%		0.000%		8	0	0
D:473C--473F	vpint	never	0.000%		0.000%		4	0	0
D:4740--4743	vppuint	never	0.000%		0.000%		4	0	0
D:4744--4748	vdarray	never	0.000%		0.000%		5	0	0
D:474C--474F	vpuint	never	0.000%		0.000%		4	0	0
D:4750--4763	ast	readwrite	0.000%		100.000%		20	0	20
D:4764--47C7	pLinkedListBuf	p-write	0.000%		80.000%		100	0	80
D:47C8--47CB	vshortrecord	never	0.000%		0.000%		4	0	0
D:47CC--47CF	datas	never	0.000%		0.000%		4	0	0
D:47D0--47D1	vushort	never	0.000%		0.000%		2	0	0
D:47D4--47D7	vppuchar	never	0.000%		0.000%		4	0	0
D:47D8--4967	stra2	never	0.000%		0.000%		400	0	0
D:4968--497F	aun	never	0.000%		0.000%		24	0	0
D:4980--4997	vbfield	p-rd p-wr	0.000%		87.500%		24	0	21
D:4998--4998	vuchar	never	0.000%		0.000%		1	0	0
D:499C--499F	vpulong	never	0.000%		0.000%		4	0	0
D:49A0--49A3	vpshort	never	0.000%		0.000%		4	0	0
D:49A4--49CB	vtarray	never	0.000%		0.000%		40	0	0
D:49CC--49CF	vtdef1	never	0.000%		0.000%		4	0	0
D:49D0--49DF	str6	never	0.000%		0.000%		16	0	0
D:49E0--49E3	funcptr	readwrite	0.000%		100.000%		4	0	4
D:49E4--49E5	vshort	never	0.000%		0.000%		2	0	0
D:49E8--49EB	vppushort	never	0.000%		0.000%		4	0	0
D:49F0--5D8F	sinewave	write	0.000%		100.000%		5024	0	5024
D:5D90--5D93	vuint	never	0.000%		0.000%		4	0	0

address	tree	coverage	read	0% 100%	write	0% 100%	bytes	read	write
D:5D94--5D94	enumvar	write	0.000%		100.000%		1	0	1
D:5D98--5DAA	flags	never	0.000%		0.000%		19	0	0
D:5DAC--5DAF	vulong	never	0.000%		0.000%		4	0	0
D:5DB0--5DB3	vplong	never	0.000%		0.000%		4	0	0
D:5DB4--5DB4	vchar	readwrite	0.000%		100.000%		1	0	1
D:5DB8--5DBB	vppulong	never	0.000%		0.000%		4	0	0
D:5DBC--5DCF	str2	never	0.000%		0.000%		20	0	0
D:5DD0--5DD3	vpuchar	never	0.000%		0.000%		4	0	0
D:5DD4--5DFB	varray	never	0.000%		0.000%		40	0	0
D:5DFC--5E03	vstruct1	never	0.000%		0.000%		8	0	0
D:5E04--5ECB	stra1	never	0.000%		0.000%		200	0	0
D:5ECC--5ECF	vlong	readwrite	0.000%		100.000%		4	0	4
D:5ED0--5EE7	vtriplearray	p-write	0.000%		16.666%		24	0	4

7. List

Coverage	Addr/Line	Code	Label	Mnemonics	Comment/Justification
		\\sieve\crt0\crt0.sx			
		...			
	4				
	5	#ifdef __thumb__			
	6	# define subs sub			
	7	#endif			
	8				
	9				
	10	_start:			
	11	/* set stack and frame pointer */			
never	12	ldr r1, .SP			
never	ST:00000400	49 0F	_start:	ldr r1,0x440	
never	13	mov r0, #0			
never	ST:00000402	20 00		mov r0,#0x0	
never	14	mov sp, r1			
never	ST:00000404	46 8D		mov r13,r1	
never	15	mov fp, r0			
never	ST:00000406	46 83		mov r11,r0	
	16				
	17	/* clear .bss-section */			
never	18	ldr r4, .BSS_S			
never	ST:00000408	4C 0E		ldr r4,0x444	
never	19	ldr r5, .BSS_E			
never	ST:0000040A	4D 0F		ldr r5,0x448	
never	20	subs r5, r5, r4			
never	ST:0000040C	1B 2D		sub r5,r5,r4	
never	21	beq bss_done			
never	ST:0000040E	D0 04		beq 0x41A	
never	22	mov r0, #0			
never	ST:00000410	20 00		mov r0,#0x0	
	23	bss_clear:			
never	24	strb r0, [r4]			
never	ST:00000412	70 20	bss_clear:	strb r0,[r4]	
never	25	add r4, r4, #1			
never	ST:00000414	34 01		add r4,#0x1	
never	26	subs r5, r5, #1			
never	ST:00000416	3D 01		sub r5,#0x1	
never	27	bne bss_clear			
never	ST:00000418	D1 FB		bne 0x412	
		...			
	32	init_data:			
	33	ldr r0, .DATA_S			
	34	ldr r1, .DATA_LADDR			
	35	ldr r2, .DATA_SIZE			
	36	bl memcpy			
	37	#endif			
	38				
	39	/* initialization of e.g. static objects */			
never	40	ldr r4, .INIT_S			
never	ST:0000041A	4C 0C	bss_done:	ldr r4,0x44C	
never	41	ldr r5, .INIT_E			
never	ST:0000041C	4D 0C		ldr r5,0x450	
never	42	subs r5, r5, r4			
never	ST:0000041E	1B 2D		sub r5,r5,r4	
never	43	beq init_done			
never	ST:00000420	D0 09		beq 0x436	
	44	init:			
never	45	ldr r3,[r4]			
never	ST:00000422	68 23	init:	ldr r3,[r4]	
never	46	push {r0,r5}			
never	ST:00000424	B4 21		push {r0,r5}	
	47	#ifdef __thumb__			
never	48	mov r1, pc			
never	ST:00000426	46 79		mov r1,pc	
never	49	add r1, #5			
never	ST:00000428	31 05		add r1,#0x5	

Coverage	Addr/Line	Code	Label	Mnemonics	Comment/Justification
never		mov lr, r1			
never	ST:0000042A	46 8E		mov r14,r1	
	51	#else			
	52	mov lr,pc			
	53	#endif			
never	54	bx r3			
never	ST:0000042C	47 18		bx r3	
never	55	pop {r0,r5}			
never	ST:0000042E	BC 21		pop {r0,r5}	
never	56	add r4, r4, #4			
never	ST:00000430	34 04		add r4,#0x4	
never	57	subs r5, r5, #4			
never	ST:00000432	3D 04		sub r5,#0x4	
never	58	bne init			
never	ST:00000434	D1 F5		bne 0x422	
	59	init_done:			
	60				
	61	gomain:			
never	62	mov r0, #0			
never	ST:00000436	20 00	gomain:	mov r0,#0x0	
never	63	mov r1, #0			
never	ST:00000438	21 00		mov r1,#0x0	
never	64	bl main			
never	ST:0000043A	F0 00 FF 17		bl 0x126C	
	65				
	66	_exit:			
never	67	b _exit			
never	ST:0000043E	E7 FE	_exit:	b 0x43E	
never	SP:00000440	00 00 80 00		dcd 0x8000	
never	SP:00000444	00 00 46 B4		dcd 0x46B4	
never	SP:00000448	00 00 5E E8		dcd 0x5EE8	
never	SP:0000044C	00 00 46 20		dcd 0x4620	
never	SP:00000450	00 00 46 20		dcd 0x4620	
never	SP:00000454	00 00 46 20		dcd 0x4620	
never	SP:00000458	00 00 46 20		dcd 0x4620	
never	SP:0000045C	00 00 00 94		dcd 0x94	
		\\sieve\isr\isr.c			
	1	extern void Monitor_Handler(void);			
	2				
	3				
	4	/* Interrupt Service Routine for Timer0 to serve monitor code			
	5	* for real-time memory access or			
	6	* on an ARM922T Excalibur device			
	7	*/			
	8	void exc_isr() __attribute__ ((interrupt ("IRQ")));			
	9	void exc_isr()			
never	10	{			
never	SR:00000460	E2 4E E0 04	exc_isr:	sub r14,r14,#0x4	
never	SR:00000464	E9 2D 58 1F		stmdb r13!,{r0-r4,r11-r12,r14}	
never	SR:00000468	E2 8D B0 1C		add r11,r13,#0x1C	
never	11	Monitor_Handler();			
never	SR:0000046C	EB 00 05 BB		bl 0x1B60	
never	12	*((unsigned int*)0x7FFFC200)=0x1C; // remove interrrupt from Timer0			
never	SR:00000470	E5 9F 30 0C		ldr r3,0x484	
never	SR:00000474	E3 A0 20 1C		mov r2,#0x1C	
never	SR:00000478	E5 83 20 00		str r2,[r3]	
never	13	}			
never	SR:0000047C	E2 4B D0 1C		sub r13,r11,#0x1C	
never	SR:00000480	E8 FD 98 1F		ldmia r13!,{r0-r4,r11-r12,pc}^	
never	SP:00000484	7F FF C2 00		dcd 0x7FFFC200	
	14				
	15				
	16	/* Interrupt Service Routine for Timer0 to serve monitor code			
	17	* for real-time snooping of the PC			
	18	* on an ARM922T Excalibur device			
	19	*/			
	20	void exc_snoop() __attribute__ ((interrupt ("IRQ")));			
	21	void exc_snoop()			
never	22	{			
never	SR:00000488	E9 2D 08 0C	exc_snoop:	stmdb r13!,{r2-r3,r11}	

Coverage	Addr/Line	Code	Label	Mnemonics	Comment/Justification
never	SR:0000048C	E2 8D B0 08		add r11,r13,#0x8	
never	23	__asm__("mcr p14,0,r14,c1,c0"); // save location at which interrupt occured to DCC register			
never	SR:00000490	EE 01 EE 10		mcr p14,0x0,r14,c1,c0,0x0	
never	24	*((unsigned int*)0x7FFFC200)=0x1C; // remove interrrupt from Timer0			
never	SR:00000494	E5 9F 30 10		ldr r3,0x4AC	
never	SR:00000498	E3 A0 20 1C		mov r2,#0x1C	
never	SR:0000049C	E5 83 20 00		str r2,[r3]	
never	25	}			
never	SR:000004A0	E2 4B D0 08		sub r13,r11,#0x8	
never	SR:000004A4	E8 BD 08 0C		ldmia r13!,{r2-r3,r11}	
never	SR:000004A8	E2 5E F0 04		subs pc,r14,#0x4	
never	SP:000004AC	7F FF C2 00		dcd 0x7FFFC200	
		\\sieve\sieve\sieve.c			
		...			
	136	struct abc z4;			
	137	} vunion;			
	138				
	139	enum enumtyp { enum1, enum2, enum4=4, enum7=7, enum8, enumx= -1 } enumvar;			
	140				
	141	int (*funcptr) (void);			
	142				
	143	void func0(void) /* empty function */			
never	144	{			
never	ST:000004B0	B5 80	func0:	push {r7,r14}	
never	ST:000004B2	AF 00		add r7,sp,#0x0	
never	145	}			
never	ST:000004B4	46 BD		mov r13,r7	
never	ST:000004B6	BC 80		pop {r7}	
never	ST:000004B8	BC 01		pop {r0}	
never	ST:000004BA	47 00		bx r0	
	146				
	147	static void func1(int * intptr) /* static function */			
ok	148	{			
ok	ST:000004BC	B5 80	func1:	push {r7,r14}	
ok	ST:000004BE	B0 82		sub sp,#0x8	
ok	ST:000004C0	AF 00		add r7,sp,#0x0	
ok	ST:000004C2	60 78		str r0,[r7,#0x4]	
ok	149	(*intptr)++;			
ok	ST:000004C4	68 7B		ldr r3,[r7,#0x4]	
ok	ST:000004C6	68 1B		ldr r3,[r3]	
ok	ST:000004C8	1C 5A		add r2,r3,#0x1	
ok	ST:000004CA	68 7B		ldr r3,[r7,#0x4]	
ok	ST:000004CC	60 1A		str r2,[r3]	
ok	150	}			
ok	ST:000004CE	46 BD		mov r13,r7	
ok	ST:000004D0	B0 02		add sp,#0x8	
ok	ST:000004D2	BC 80		pop {r7}	
ok	ST:000004D4	BC 01		pop {r0}	
ok	ST:000004D6	47 00		bx r0	
	151				
	152	void func2(void)			
ok	153	{			
ok	ST:000004D8	B5 90	func2:	push {r4,r7,r14}	
ok	ST:000004DA	B0 83		sub sp,#0x0C	
ok	ST:000004DC	AF 00		add r7,sp,#0x0	
	154	int autovar;			
	155	register int regvar;			
	156	static int fstatic = 44; /* initialized static variable */			
	157	static int fstatic2; /* not initialized static variable */			
	158				
ok	159	autovar = regvar = fstatic;			
ok	ST:000004DE	4B 1C		ldr r3,0x550	
ok	ST:000004E0	68 1C		ldr r4,[r3]	
ok	ST:000004E2	60 7C		str r4,[r7,#0x4]	
ok	160	autovar++;			
ok	ST:000004E4	68 7B		ldr r3,[r7,#0x4]	
ok	ST:000004E6	33 01		add r3,#0x1	
ok	ST:000004E8	60 7B		str r3,[r7,#0x4]	
	161				

Coverage	Addr/Line	Code	Label	Mnemonics		Comment/Justification
ok	162	*func1(&autovar); /* to force autovar as stack-scope */*				
ok	ST:000004EA	1D 3B		add	r3,r7,#0x4	
ok	ST:000004EC	1C 18		mov	r0,r3	
ok	ST:000004EE	F7 FF FF E5		bl	0x4BC	
ok	163	*func1(&fstatic); /* to force fstatic as static-scope */*				
ok	ST:000004F2	4B 17		ldr	r3,0x550	
ok	ST:000004F4	1C 18		mov	r0,r3	
ok	ST:000004F6	F7 FF FF E1		bl	0x4BC	
	164					
ok	165	*for (regvar = 0; regvar < 5 ; regvar++)*				
ok	ST:000004FA	24 00		mov	r4,#0x0	
ok	ST:000004FC	E0 08		b	0x510	
ok	166	*mstatic1 += regvar*autovar;*				
ok	ST:000004FE	68 7B		ldr	r3,[r7,#0x4]	
ok	ST:00000500	1C 1A		mov	r2,r3	
ok	ST:00000502	43 62		mul	r2,r4	
ok	ST:00000504	4B 13		ldr	r3,0x554	
ok	ST:00000506	68 1B		ldr	r3,[r3]	
ok	ST:00000508	18 D2		add	r2,r2,r3	
ok	ST:0000050A	4B 12		ldr	r3,0x554	
ok	ST:0000050C	60 1A		str	r2,[r3]	
	164					
ok	165	*for (regvar = 0; regvar < 5 ; regvar++)*				
ok	ST:0000050E	34 01		add	r4,#0x1	
ok	ST:00000510	2C 04		cmp	r4,#0x4	
ok	ST:00000512	DD F4		ble	0x4FE	
	167					
ok	168	*fstatic += mstatic1;*				
ok	ST:00000514	4B 0E		ldr	r3,0x550	
ok	ST:00000516	68 1A		ldr	r2,[r3]	
ok	ST:00000518	4B 0E		ldr	r3,0x554	
ok	ST:0000051A	68 1B		ldr	r3,[r3]	
ok	ST:0000051C	18 D2		add	r2,r2,r3	
ok	ST:0000051E	4B 0C		ldr	r3,0x550	
ok	ST:00000520	60 1A		str	r2,[r3]	
ok	169	*fstatic2 = 2*fstatic;*				
ok	ST:00000522	4B 0B		ldr	r3,0x550	
ok	ST:00000524	68 1B		ldr	r3,[r3]	
ok	ST:00000526	00 5A		lsl	r2,r3,#0x1	
ok	ST:00000528	4B 0B		ldr	r3,0x558	
ok	ST:0000052A	60 1A		str	r2,[r3]	
	170					
taken	171	*if (mcount < 500)*				
ok	ST:0000052C	4B 0B		ldr	r3,0x55C	
ok	ST:0000052E	68 1A		ldr	r2,[r3]	
ok	ST:00000530	4B 0B		ldr	r3,0x560	
ok	ST:00000532	42 9A		cmp	r2,r3	
taken	ST:00000534	DD 05		ble	0x542	
	173					
never	174	*func1(&fstatic2);*				
never	ST:00000536	4B 08		ldr	r3,0x558	
never	ST:00000538	1C 18		mov	r0,r3	
never	ST:0000053A	F7 FF FF BF		bl	0x4BC	
	175					
never	176	*return;*				
never	ST:0000053E	46 C0		nop		
never	ST:00000540	E0 00		b	0x544	
ok	172	*return;*				
ok	ST:00000542	46 C0		nop		
partial	177	*}*				
ok	ST:00000544	46 BD		mov	r13,r7	
ok	ST:00000546	B0 03		add	sp,#0x0C	
ok	ST:00000548	BC 90		pop	{r4,r7}	
ok	ST:0000054A	BC 01		pop	{r0}	
ok	ST:0000054C	47 00		bx	r0	
never	ST:0000054E	46 C0		nop		
read	SP:00000550	00 00 46 5C		dcd	0x465C	
read	SP:00000554	00 00 46 C0		dcd	0x46C0	

Coverage	Addr/Line	Code	Label	Mnemonics		Comment/Justification
read	SP:00000558	00 00 46 E0		dcd	0x46E0	
read	SP:0000055C	00 00 46 C8		dcd	0x46C8	
read	SP:00000560	00 00 01 F3		dcd	0x1F3	
	178					
	179	*void func2a(void)*				
ok	180	*{*				
ok	ST:00000564	B5 90	func2a:	push	{r4,r7,r14}	
ok	ST:00000566	B0 83		sub	sp,#0x0C	
ok	ST:00000568	AF 00		add	r7,sp,#0x0	
	181	*char autovar; /* char stack variable */*				
	182	*register char regvar; /* char register variable */*				
	183					
ok	184	*autovar = regvar = (char) mstatic1;*				
ok	ST:0000056A	4B 13		ldr	r3,0x5B8	
ok	ST:0000056C	68 1B		ldr	r3,[r3]	
ok	ST:0000056E	06 1B		lsl	r3,r3,#0x18	
ok	ST:00000570	0E 1C		lsr	r4,r3,#0x18	
ok	ST:00000572	1D FB		add	r3,r7,#0x7	
ok	ST:00000574	1C 22		mov	r2,r4	
ok	ST:00000576	70 1A		strb	r2,[r3]	
ok	185	*autovar++;*				
ok	ST:00000578	1D FB		add	r3,r7,#0x7	
ok	ST:0000057A	1D FA		add	r2,r7,#0x7	
ok	ST:0000057C	78 12		ldrb	r2,[r2]	
ok	ST:0000057E	32 01		add	r2,#0x1	
ok	ST:00000580	70 1A		strb	r2,[r3]	
	186					
ok	187	*for (regvar = 0; regvar < (char) 5 ; regvar++)*				
ok	ST:00000582	24 00		mov	r4,#0x0	
ok	ST:00000584	E0 10		b	0x5A8	
ok	188	*vchar += regvar*autovar;*				
ok	ST:00000586	1D FB		add	r3,r7,#0x7	
ok	ST:00000588	78 1B		ldrb	r3,[r3]	
ok	ST:0000058A	1C 1A		mov	r2,r3	
ok	ST:0000058C	1C 13		mov	r3,r2	
ok	ST:0000058E	43 63		mul	r3,r4	
ok	ST:00000590	06 1B		lsl	r3,r3,#0x18	
ok	ST:00000592	0E 1A		lsr	r2,r3,#0x18	
ok	ST:00000594	4B 09		ldr	r3,0x5BC	
ok	ST:00000596	78 1B		ldrb	r3,[r3]	
ok	ST:00000598	18 D3		add	r3,r2,r3	
ok	ST:0000059A	06 1B		lsl	r3,r3,#0x18	
ok	ST:0000059C	0E 1A		lsr	r2,r3,#0x18	
ok	ST:0000059E	4B 07		ldr	r3,0x5BC	
ok	ST:000005A0	70 1A		strb	r2,[r3]	
	186					
ok	187	*for (regvar = 0; regvar < (char) 5 ; regvar++)*				
ok	ST:000005A2	1C 63		add	r3,r4,#0x1	
ok	ST:000005A4	06 1B		lsl	r3,r3,#0x18	
ok	ST:000005A6	0E 1C		lsr	r4,r3,#0x18	
ok	ST:000005A8	2C 04		cmp	r4,#0x4	
ok	ST:000005AA	D9 EC		bls	0x586	
partial	189	*}*				
ok	ST:000005AC	46 BD		mov	r13,r7	
ok	ST:000005AE	B0 03		add	sp,#0x0C	
ok	ST:000005B0	BC 90		pop	{r4,r7}	
ok	ST:000005B2	BC 01		pop	{r0}	
ok	ST:000005B4	47 00		bx	r0	
never	ST:000005B6	46 C0		nop		
read	SP:000005B8	00 00 46 C0		dcd	0x46C0	
read	SP:000005BC	00 00 5D B4		dcd	0x5DB4	
	190					
	191	*void func2b(void)*				
ok	192	*{*				
ok	ST:000005C0	B5 90	func2b:	push	{r4,r7,r14}	
ok	ST:000005C2	B0 83		sub	sp,#0x0C	
ok	ST:000005C4	AF 00		add	r7,sp,#0x0	
	193	*long autovar; /* long stack variable */*				

Coverage	Addr/Line	Code	Label	Mnemonics	Comment/Justification
	194	*register long regvar; /* long register variable */*			
	195				
ok	196	*autovar = regvar = mstatic1;*			
ok	ST:000005C6	4B 0C		ldr r3,0x5F8	
ok	ST:000005C8	68 1C		ldr r4,[r3]	
ok	ST:000005CA	60 7C		str r4,[r7,#0x4]	
ok	197	*autovar++;*			
ok	ST:000005CC	68 7B		ldr r3,[r7,#0x4]	
ok	ST:000005CE	33 01		add r3,#0x1	
ok	ST:000005D0	60 7B		str r3,[r7,#0x4]	
	198				
ok	199	*for (regvar = 0; regvar < 5l ; regvar++)*			
ok	ST:000005D2	24 00		mov r4,#0x0	
ok	ST:000005D4	E0 08		b 0x5E8	
ok	200	*vlong += regvar*autovar;*			
ok	ST:000005D6	68 7B		ldr r3,[r7,#0x4]	
ok	ST:000005D8	1C 1A		mov r2,r3	
ok	ST:000005DA	43 62		mul r2,r4	
ok	ST:000005DC	4B 07		ldr r3,0x5FC	
ok	ST:000005DE	68 1B		ldr r3,[r3]	
ok	ST:000005E0	18 D2		add r2,r2,r3	
ok	ST:000005E2	4B 06		ldr r3,0x5FC	
ok	ST:000005E4	60 1A		str r2,[r3]	
	198				
ok	199	*for (regvar = 0; regvar < 5l ; regvar++)*			
ok	ST:000005E6	34 01		add r4,#0x1	
ok	ST:000005E8	2C 04		cmp r4,#0x4	
ok	ST:000005EA	DD F4		ble 0x5D6	
partial	201	*}*			
ok	ST:000005EC	46 BD		mov r13,r7	
ok	ST:000005EE	B0 03		add sp,#0x0C	
ok	ST:000005F0	BC 90		pop {r4,r7}	
ok	ST:000005F2	BC 01		pop {r0}	
ok	ST:000005F4	47 00		bx r0	
never	ST:000005F6	46 C0		nop	
read	SP:000005F8	00 00 46 C0		dcd 0x46C0	
read	SP:000005FC	00 00 5E CC		dcd 0x5ECC	
	202				
	203	*void func2c(void)*			
ok	204	*{*			
ok	ST:00000600	B5 F0	func2c:	push {r4-r7,r14}	
ok	ST:00000602	B0 83		sub sp,#0x0C	
ok	ST:00000604	AF 00		add r7,sp,#0x0	
	205	*double autovar; /* double stack variable */*			
	206	*register double regvar; /* double register variable */*			
	207				
ok	208	*autovar = regvar = (double) mstatic1;*			
ok	ST:00000606	4B 2C		ldr r3,0x6B8	
ok	ST:00000608	68 1B		ldr r3,[r3]	
ok	ST:0000060A	1C 18		mov r0,r3	
ok	ST:0000060C	F0 03 FE 94		bl 0x4338	
ok	ST:00000610	1C 03		mov r3,r0	
ok	ST:00000612	1C 0C		mov r4,r1	
ok	ST:00000614	1C 25		mov r5,r4	
ok	ST:00000616	1C 1C		mov r4,r3	
ok	ST:00000618	60 3C		str r4,[r7]	
ok	ST:0000061A	60 7D		str r5,[r7,#0x4]	
ok	209	*autovar++;*			
ok	ST:0000061C	68 38		ldr r0,[r7]	
ok	ST:0000061E	68 79		ldr r1,[r7,#0x4]	
ok	ST:00000620	4B 20		ldr r3,0x6A4	
ok	ST:00000622	4A 1F		ldr r2,0x6A0	
ok	ST:00000624	F0 03 FE 8C		bl 0x4340	
ok	ST:00000628	1C 03		mov r3,r0	
ok	ST:0000062A	1C 0C		mov r4,r1	
ok	ST:0000062C	60 3B		str r3,[r7]	
ok	ST:0000062E	60 7C		str r4,[r7,#0x4]	
	210				

Coverage	Addr/Line	Code	Label	Mnemonics		Comment/Justification
ok	211	*for (regvar = 0.0; regvar < 5.0 ; regvar += 1.0)*				
ok	ST:00000630	4C 1D		ldr	r4,0x6A8	
ok	ST:00000632	4D 1E		ldr	r5,0x6AC	
ok	ST:00000634	E0 1D		b	0x672	
ok	212	*vdouble += regvar*autovar;*				
ok	ST:00000636	1C 20		mov	r0,r4	
ok	ST:00000638	1C 29		mov	r1,r5	
ok	ST:0000063A	68 3A		ldr	r2,[r7]	
ok	ST:0000063C	68 7B		ldr	r3,[r7,#0x4]	
ok	ST:0000063E	F0 03 FE 6B		bl	0x4318	
ok	ST:00000642	1C 02		mov	r2,r0	
ok	ST:00000644	1C 0B		mov	r3,r1	
ok	ST:00000646	1C 10		mov	r0,r2	
ok	ST:00000648	1C 19		mov	r1,r3	
ok	ST:0000064A	4B 1C		ldr	r3,0x6BC	
ok	ST:0000064C	68 1A		ldr	r2,[r3]	
ok	ST:0000064E	68 5B		ldr	r3,[r3,#0x4]	
ok	ST:00000650	F0 03 FE 76		bl	0x4340	
ok	ST:00000654	1C 02		mov	r2,r0	
ok	ST:00000656	1C 0B		mov	r3,r1	
ok	ST:00000658	49 18		ldr	r1,0x6BC	
ok	ST:0000065A	60 0A		str	r2,[r1]	
ok	ST:0000065C	60 4B		str	r3,[r1,#0x4]	
	210					
ok	211	*for (regvar = 0.0; regvar < 5.0 ; regvar += 1.0)*				
ok	ST:0000065E	1C 20		mov	r0,r4	
ok	ST:00000660	1C 29		mov	r1,r5	
ok	ST:00000662	4B 10		ldr	r3,0x6A4	
ok	ST:00000664	4A 0E		ldr	r2,0x6A0	
ok	ST:00000666	F0 03 FE 6B		bl	0x4340	
ok	ST:0000066A	1C 03		mov	r3,r0	
ok	ST:0000066C	1C 0C		mov	r4,r1	
ok	ST:0000066E	1C 25		mov	r5,r4	
ok	ST:00000670	1C 1C		mov	r4,r3	
ok	ST:00000672	23 01		mov	r3,#0x1	
ok	ST:00000674	1C 1E		mov	r6,r3	
ok	ST:00000676	1C 20		mov	r0,r4	
ok	ST:00000678	1C 29		mov	r1,r5	
ok	ST:0000067A	4A 0D		ldr	r2,0x6B0	
ok	ST:0000067C	4B 0D		ldr	r3,0x6B4	
ok	ST:0000067E	F0 03 FE 47		bl	0x4310	
ok	ST:00000682	1C 03		mov	r3,r0	
ok	ST:00000684	D1 01		bne	0x68A	
ok	ST:00000686	23 00		mov	r3,#0x0	
ok	ST:00000688	1C 1E		mov	r6,r3	
ok	ST:0000068A	06 33		lsl	r3,r6,#0x18	
ok	ST:0000068C	0E 1B		lsr	r3,r3,#0x18	
ok	ST:0000068E	D1 D2		bne	0x636	
partial	213	*}*				
ok	ST:00000690	46 BD		mov	r13,r7	
ok	ST:00000692	B0 03		add	sp,#0x0C	
ok	ST:00000694	BC F0		pop	{r4-r7}	
ok	ST:00000696	BC 01		pop	{r0}	
ok	ST:00000698	47 00		bx	r0	
never	ST:0000069A	46 C0		nop		
never	ST:0000069C	46 C0		nop		
never	ST:0000069E	46 C0		nop		
read	SP:000006A0	00 00 00 00		dcd	0x0	
read	SP:000006A4	3F F0 00 00		dcd	0x3FF00000	
read	SP:000006A8	00 00 00 00		dcd	0x0	
read	SP:000006AC	00 00 00 00		dcd	0x0	
read	SP:000006B0	00 00 00 00		dcd	0x0	
read	SP:000006B4	40 14 00 00		dcd	0x40140000	
read	SP:000006B8	00 00 46 C0		dcd	0x46C0	
read	SP:000006BC	00 00 46 38		dcd	0x4638	
	214					
	215	*void func2d(void)*				
ok	216	*{*				

Coverage	Address	Code	Label	Mnemonics		Comment/Justification
ok	ST:000006C0	B5 80	func2d:	push	{r4,r7,r14}	
ok	ST:000006C2	B0 83		sub	sp,#0x0C	
ok	ST:000006C4	AF 00		add	r7,sp,#0x0	
	217	*short autovar; /* short stack variable */*				
	218	*register short regvar; /* short register variable */*				
	219					
ok	220	*autovar = regvar = mstatic1;*				
ok	ST:000006C6	4B 13		ldr	r3,0x714	
ok	ST:000006C8	68 1B		ldr	r3,[r3]	
ok	ST:000006CA	04 1B		lsl	r3,r3,#0x10	
ok	ST:000006CC	0C 1C		lsr	r4,r3,#0x10	
ok	ST:000006CE	1D BB		add	r3,r7,#0x6	
ok	ST:000006D0	1C 22		mov	r2,r4	
ok	ST:000006D2	80 1A		strh	r2,[r3]	
ok	221	*autovar++;*				
ok	ST:000006D4	1D BB		add	r3,r7,#0x6	
ok	ST:000006D6	1D BA		add	r2,r7,#0x6	
ok	ST:000006D8	88 12		ldrh	r2,[r2]	
ok	ST:000006DA	32 01		add	r2,#0x1	
ok	ST:000006DC	80 1A		strh	r2,[r3]	
	222					
ok	223	*for (regvar = 0; regvar < 5l ; regvar++)*				
ok	ST:000006DE	24 00		mov	r4,#0x0	
ok	ST:000006E0	E0 0E		b	0x700	
ok	224	*vlong += regvar*autovar;*				
ok	ST:000006E2	04 23		lsl	r3,r4,#0x10	
ok	ST:000006E4	14 1B		asr	r3,r3,#0x10	
ok	ST:000006E6	1D BA		add	r2,r7,#0x6	
ok	ST:000006E8	21 00		mov	r1,#0x0	
ok	ST:000006EA	5E 52		ldrsh	r2,[r2,r1]	
ok	ST:000006EC	43 5A		mul	r2,r3	
ok	ST:000006EE	4B 0A		ldr	r3,0x718	
ok	ST:000006F0	68 1B		ldr	r3,[r3]	
ok	ST:000006F2	18 D2		add	r2,r2,r3	
ok	ST:000006F4	4B 08		ldr	r3,0x718	
ok	ST:000006F6	60 1A		str	r2,[r3]	
	222					
ok	223	*for (regvar = 0; regvar < 5l ; regvar++)*				
ok	ST:000006F8	1C 23		mov	r3,r4	
ok	ST:000006FA	33 01		add	r3,#0x1	
ok	ST:000006FC	04 1B		lsl	r3,r3,#0x10	
ok	ST:000006FE	0C 1C		lsr	r4,r3,#0x10	
ok	ST:00000700	04 23		lsl	r3,r4,#0x10	
ok	ST:00000702	14 1B		asr	r3,r3,#0x10	
ok	ST:00000704	2B 04		cmp	r3,#0x4	
ok	ST:00000706	DD EC		ble	0x6E2	
partial	225	*}*				
ok	ST:00000708	46 BD		mov	r13,r7	
ok	ST:0000070A	B0 03		add	sp,#0x0C	
ok	ST:0000070C	BC 90		pop	{r4,r7}	
ok	ST:0000070E	BC 01		pop	{r0}	
ok	ST:00000710	47 00		bx	r0	
never	ST:00000712	46 C0		nop		
read	SP:00000714	00 00 46 C0		dcd	0x46C0	
read	SP:00000718	00 00 5E CC		dcd	0x5ECC	
	226					
	227					
	228	*static int func3(void) /* simple function */*				
ok	229	*{*				
ok	ST:0000071C	B5 80	func3:	push	{r7,r14}	
ok	ST:0000071E	AF 00		add	r7,sp,#0x0	
ok	230	*return 5;*				
ok	ST:00000720	23 05		mov	r3,#0x5	
ok	231	*}*				
ok	ST:00000722	1C 18		mov	r0,r3	
ok	ST:00000724	46 BD		mov	r13,r7	
ok	ST:00000726	BC 80		pop	{r7}	
ok	ST:00000728	BC 02		pop	{r1}	
	ST:0000072A			bx	r1	

Coverage	Addr/Line	Code	Label	Mnemonics	Comment/Justification
	232	47 08			
	233	*struct struct1 func4(struct struct1 str) /* function returning struct */*			
	234	*{*			
ok	ST:0000072C	B0 84	func4:	sub	sp,#0x10
ok	ST:0000072E	B5 90		push	{r4,r7,r14}
ok	ST:00000730	B0 83		sub	sp,#0x0C
ok	ST:00000732	AF 00		add	r7,sp,#0x0
ok	ST:00000734	60 78		str	r0,[r7,#0x4]
ok	ST:00000736	20 04		mov	r0,#0x4
ok	ST:00000738	24 18		mov	r4,#0x18
ok	ST:0000073A	19 E4		add	r4,r4,r7
ok	ST:0000073C	18 20		add	r0,r4,r0
ok	ST:0000073E	60 01		str	r1,[r0]
ok	ST:00000740	60 42		str	r2,[r0,#0x4]
ok	ST:00000742	60 83		str	r3,[r0,#0x8]
ok	235	*str.count++;*			
ok	ST:00000744	23 04		mov	r3,#0x4
ok	ST:00000746	20 18		mov	r0,#0x18
ok	ST:00000748	19 C0		add	r0,r0,r7
ok	ST:0000074A	18 C3		add	r3,r0,r3
ok	ST:0000074C	68 5B		ldr	r3,[r3,#0x4]
ok	ST:0000074E	1C 5A		add	r2,r3,#0x1
ok	ST:00000750	23 04		mov	r3,#0x4
ok	ST:00000752	21 18		mov	r1,#0x18
ok	ST:00000754	19 C9		add	r1,r1,r7
ok	ST:00000756	18 CB		add	r3,r1,r3
ok	ST:00000758	60 5A		str	r2,[r3,#0x4]
	236				
ok	237	*return str;*			
ok	ST:0000075A	68 7B		ldr	r3,[r7,#0x4]
ok	ST:0000075C	22 04		mov	r2,#0x4
ok	ST:0000075E	24 18		mov	r4,#0x18
ok	ST:00000760	19 E4		add	r4,r4,r7
ok	ST:00000762	18 A2		add	r2,r4,r2
ok	ST:00000764	CA 13		ldmia	r2!,{r0-r1,r4}
ok	ST:00000766	C3 13		stmia	r3!,{r0-r1,r4}
ok	ST:00000768	CA 03		ldmia	r2!,{r0-r1}
ok	ST:0000076A	C3 03		stmia	r3!,{r0-r1}
partial	238	*}*			
ok	ST:0000076C	68 78		ldr	r0,[r7,#0x4]
ok	ST:0000076E	46 BD		mov	r13,r7
ok	ST:00000770	B0 03		add	sp,#0x0C
ok	ST:00000772	BC 90		pop	{r4,r7}
ok	ST:00000774	BC 08		pop	{r3}
ok	ST:00000776	B0 04		add	sp,#0x10
ok	ST:00000778	47 18		bx	r3
never	ST:0000077A	46 C0		nop	padding NOP
	239				
	240	*int func5(int a, char b, long c) /* multiple arguments */*			
ok	241	*{*			
ok	ST:0000077C	B5 80	func5:	push	{r7,r14}
ok	ST:0000077E	B0 84		sub	sp,#0x10
ok	ST:00000780	AF 00		add	r7,sp,#0x0
ok	ST:00000782	60 F8		str	r0,[r7,#0x0C]
ok	ST:00000784	60 7A		str	r2,[r7,#0x4]
ok	ST:00000786	1C 3B		mov	r3,r7
ok	ST:00000788	33 0B		add	r3,#0x0B
ok	ST:0000078A	1C 0A		mov	r2,r1
ok	ST:0000078C	70 1A		strb	r2,[r3]
ok	242	*return a+b*c;*			
ok	ST:0000078E	1C 3B		mov	r3,r7
ok	ST:00000790	33 0B		add	r3,#0x0B
ok	ST:00000792	78 1B		ldrb	r3,[r3]
ok	ST:00000794	68 7A		ldr	r2,[r7,#0x4]
ok	ST:00000796	43 5A		mul	r2,r3
ok	ST:00000798	68 FB		ldr	r3,[r7,#0x0C]
ok	ST:0000079A	18 D3		add	r3,r2,r3

Coverage	Addr/Line	Code	Label	Mnemonics	Comment/Justification
ok	ST:0000079C	1C 18		mov r0,r3	
ok	ST:0000079E	46 BD		mov r13,r7	
ok	ST:000007A0	B0 04		add sp,#0x10	
ok	ST:000007A2	BC 80		pop {r7}	
ok	ST:000007A4	BC 02		pop {r1}	
ok	ST:000007A6	47 08		bx r1	
	244				
	245	*float func6(float a, float b)*			
ok	246	*{*			
ok	ST:000007A8	B5 80	func6:	push {r7,r14}	
ok	ST:000007AA	B0 82		sub sp,#0x8	
ok	ST:000007AC	AF 00		add r7,sp,#0x0	
ok	ST:000007AE	60 78		str r0,[r7,#0x4]	
ok	ST:000007B0	60 39		str r1,[r7]	
ok	247	*vfloat = 1.0;*			
ok	ST:000007B2	4B 0B		ldr r3,0x7E0	
ok	ST:000007B4	4A 0B		ldr r2,0x7E4	
ok	ST:000007B6	60 1A		str r2,[r3]	
ok	248	*vfloat = -1.0;*			
ok	ST:000007B8	4B 09		ldr r3,0x7E0	
ok	ST:000007BA	4A 0B		ldr r2,0x7E8	
ok	ST:000007BC	60 1A		str r2,[r3]	
ok	249	*vfloat = 10.0;*			
ok	ST:000007BE	4B 08		ldr r3,0x7E0	
ok	ST:000007C0	4A 0A		ldr r2,0x7EC	
ok	ST:000007C2	60 1A		str r2,[r3]	
ok	250	*vfloat = 1.6;*			
ok	ST:000007C4	4B 06		ldr r3,0x7E0	
ok	ST:000007C6	4A 0A		ldr r2,0x7F0	
ok	ST:000007C8	60 1A		str r2,[r3]	
	251				
ok	252	*return a*b;*			
ok	ST:000007CA	68 78		ldr r0,[r7,#0x4]	
ok	ST:000007CC	68 39		ldr r1,[r7]	
ok	ST:000007CE	F0 03 FD BB		bl 0x4348	
ok	ST:000007D2	1C 03		mov r3,r0	
partial	253	*}*			
ok	ST:000007D4	1C 18		mov r0,r3	
ok	ST:000007D6	46 BD		mov r13,r7	
ok	ST:000007D8	B0 02		add sp,#0x8	
ok	ST:000007DA	BC 80		pop {r7}	
ok	ST:000007DC	BC 02		pop {r1}	
ok	ST:000007DE	47 08		bx r1	
read	SP:000007E0	00 00 46 30		dcd 0x4630	
read	SP:000007E4	3F 80 00 00		dcd 0x3F800000	
read	SP:000007E8	BF 80 00 00		dcd 0xBF800000	
read	SP:000007EC	41 20 00 00		dcd 0x41200000	
read	SP:000007F0	3F CC CC CD		dcd 0x3FCCCCCD	
	254				
	255	*double func7(double a, double b)*			
ok	256	*{*			
ok	ST:000007F4	B5 90	func7:	push {r4,r7,r14}	
ok	ST:000007F6	B0 85		sub sp,#0x14	
ok	ST:000007F8	AF 00		add r7,sp,#0x0	
ok	ST:000007FA	60 B8		str r0,[r7,#0x8]	
ok	ST:000007FC	60 F9		str r1,[r7,#0x0C]	
ok	ST:000007FE	60 3A		str r2,[r7]	
ok	ST:00000800	60 7B		str r3,[r7,#0x4]	
ok	257	*vdouble = 1.0;*			
ok	ST:00000802	4A 19		ldr r2,0x868	
ok	ST:00000804	4C 11		ldr r4,0x84C	
ok	ST:00000806	4B 10		ldr r3,0x848	
ok	ST:00000808	60 13		str r3,[r2]	
ok	ST:0000080A	60 54		str r4,[r2,#0x4]	
ok	258	*vdouble = -1.0;*			
ok	ST:0000080C	4A 16		ldr r2,0x868	

Coverage	Address	Code	Label	Mnemonic	Comment/Justification
ok	ST:0000080E	6B 10		ldr r3,0x850	
ok	ST:00000810	4C 10		ldr r4,0x854	
ok	ST:00000812	60 13		str r3,[r2]	
ok	ST:00000814	60 54		str r4,[r2,#0x4]	
ok	259	vdouble = 10.0;			
ok	ST:00000816	4A 14		ldr r2,0x868	
ok	ST:00000818	4B 0F		ldr r3,0x858	
ok	ST:0000081A	4C 10		ldr r4,0x85C	
ok	ST:0000081C	60 13		str r3,[r2]	
ok	ST:0000081E	60 54		str r4,[r2,#0x4]	
ok	260	vdouble = 1.6;			
ok	ST:00000820	4A 11		ldr r2,0x868	
ok	ST:00000822	4B 0F		ldr r3,0x860	
ok	ST:00000824	4C 0F		ldr r4,0x864	
ok	ST:00000826	60 13		str r3,[r2]	
ok	ST:00000828	60 54		str r4,[r2,#0x4]	
	261				
ok	262	return a*b;			
ok	ST:0000082A	68 B8		ldr r0,[r7,#0x8]	
ok	ST:0000082C	68 F9		ldr r1,[r7,#0x0C]	
ok	ST:0000082E	68 3A		ldr r2,[r7]	
ok	ST:00000830	68 7B		ldr r3,[r7,#0x4]	
ok	ST:00000832	F0 03 FD 71		bl 0x4318	
ok	ST:00000836	1C 03		mov r3,r0	
ok	ST:00000838	1C 0C		mov r4,r1	
partial	263	}			
ok	ST:0000083A	1C 18		mov r0,r3	
ok	ST:0000083C	1C 21		mov r1,r4	
ok	ST:0000083E	46 BD		mov r13,r7	
ok	ST:00000840	B0 05		add sp,#0x14	
ok	ST:00000842	BC 90		pop {r4,r7}	
ok	ST:00000844	BC 04		pop {r2}	
ok	ST:00000846	47 10		bx r2	
read	SP:00000848	00 00 00 00		dcd 0x0	
read	SP:0000084C	3F F0 00 00		dcd 0x3FF00000	
read	SP:00000850	00 00 00 00		dcd 0x0	
read	SP:00000854	BF F0 00 00		dcd 0xBFF00000	
read	SP:00000858	00 00 00 00		dcd 0x0	
read	SP:0000085C	40 24 00 00		dcd 0x40240000	
read	SP:00000860	99 99 99 9A		dcd 0x9999999A	
read	SP:00000864	3F F9 99 99		dcd 0x3FF99999	
read	SP:00000868	00 00 46 38		dcd 0x4638	
		...			
	282	int p:15;			
	283	char q:6;			
	284	char r:1;			
	285	};			
	286				
	287	struct bfield vbfield;			
	288				
	289	void func8(void)			
ok	290	{			
ok	ST:0000086C	B5 80	func8:	push {r7,r14}	
ok	ST:0000086E	AF 00		add r7,sp,#0x0	
ok	291	vbfield.a = 1;			
ok	ST:00000870	4B 68		ldr r3,0xA14	
ok	ST:00000872	78 1A		ldrb r2,[r3]	
ok	ST:00000874	21 01		mov r1,#0x1	
ok	ST:00000876	43 0A		orr r2,r1	
ok	ST:00000878	70 1A		strb r2,[r3]	
ok	292	vbfield.b = 1;			
ok	ST:0000087A	4B 66		ldr r3,0xA14	
ok	ST:0000087C	78 1A		ldrb r2,[r3]	
ok	ST:0000087E	21 06		mov r1,#0x6	
ok	ST:00000880	43 8A		bic r2,r1	
ok	ST:00000882	21 02		mov r1,#0x2	
ok	ST:00000884	43 0A		orr r2,r1	
ok	ST:00000886	70 1A		strb r2,[r3]	

Coverage	Addr/Line	Code	Label	Mnemonics	Comment/Justification
ok	293	*vbfield.c = 1;*			
ok	ST:00000888	4B 62		ldr r3,0xA14	
ok	ST:0000088A	78 1A		ldrb r2,[r3]	
ok	ST:0000088C	21 38		mov r1,#0x38	
ok	ST:0000088E	43 8A		bic r2,r1	
ok	ST:00000890	21 08		mov r1,#0x8	
ok	ST:00000892	43 0A		orr r2,r1	
ok	ST:00000894	70 1A		strb r2,[r3]	
ok	294	*vbfield.d = 1;*			
ok	ST:00000896	4B 5F		ldr r3,0xA14	
ok	ST:00000898	88 19		ldrh r1,[r3]	
ok	ST:0000089A	4A 5F		ldr r2,0xA18	
ok	ST:0000089C	40 0A		and r2,r1	
ok	ST:0000089E	21 40		mov r1,#0x40	
ok	ST:000008A0	43 0A		orr r2,r1	
ok	ST:000008A2	80 1A		strh r2,[r3]	
ok	295	*vbfield.e = 1;*			
ok	ST:000008A4	4B 5B		ldr r3,0xA14	
ok	ST:000008A6	68 19		ldr r1,[r3]	
ok	ST:000008A8	4A 5C		ldr r2,0xA1C	
ok	ST:000008AA	40 0A		and r2,r1	
ok	ST:000008AC	21 80		mov r1,#0x80	
ok	ST:000008AE	01 89		lsl r1,r1,#0x6	
ok	ST:000008B0	43 0A		orr r2,r1	
ok	ST:000008B2	60 1A		str r2,[r3]	
ok	296	*vbfield.f = 1;*			
ok	ST:000008B4	4B 57		ldr r3,0xA14	
ok	ST:000008B6	88 9A		ldrh r2,[r3,#0x4]	
ok	ST:000008B8	0B D2		lsr r2,r2,#0x0F	
ok	ST:000008BA	03 D2		lsl r2,r2,#0x0F	
ok	ST:000008BC	21 01		mov r1,#0x1	
ok	ST:000008BE	43 0A		orr r2,r1	
ok	ST:000008C0	80 9A		strh r2,[r3,#0x4]	
ok	297	*vbfield.g = 1;*			
ok	ST:000008C2	4B 54		ldr r3,0xA14	
ok	ST:000008C4	79 5A		ldrb r2,[r3,#0x5]	
ok	ST:000008C6	21 80		mov r1,#0x80	
ok	ST:000008C8	42 49		neg r1,r1	
ok	ST:000008CA	43 0A		orr r2,r1	
ok	ST:000008CC	71 5A		strb r2,[r3,#0x5]	
ok	298	*vbfield.h = 1;*			
ok	ST:000008CE	4B 51		ldr r3,0xA14	
ok	ST:000008D0	88 DA		ldrh r2,[r3,#0x6]	
ok	ST:000008D2	0B D2		lsr r2,r2,#0x0F	
ok	ST:000008D4	03 D2		lsl r2,r2,#0x0F	
ok	ST:000008D6	21 01		mov r1,#0x1	
ok	ST:000008D8	43 0A		orr r2,r1	
ok	ST:000008DA	80 DA		strh r2,[r3,#0x6]	
ok	299	*vbfield.i = 1;*			
ok	ST:000008DC	4B 4D		ldr r3,0xA14	
ok	ST:000008DE	79 DA		ldrb r2,[r3,#0x7]	
ok	ST:000008E0	21 80		mov r1,#0x80	
ok	ST:000008E2	42 49		neg r1,r1	
ok	ST:000008E4	43 0A		orr r2,r1	
ok	ST:000008E6	71 DA		strb r2,[r3,#0x7]	
ok	300	*vbfield.j = 1;*			
ok	ST:000008E8	4B 4A		ldr r3,0xA14	
ok	ST:000008EA	89 19		ldrh r1,[r3,#0x8]	
ok	ST:000008EC	22 00		mov r2,#0x0	
ok	ST:000008EE	40 0A		and r2,r1	
ok	ST:000008F0	21 01		mov r1,#0x1	
ok	ST:000008F2	43 0A		orr r2,r1	
ok	ST:000008F4	81 1A		strh r2,[r3,#0x8]	
ok	301	*vbfield.k = 1;*			
ok	ST:000008F6	4B 47		ldr r3,0xA14	
ok	ST:000008F8	7A 9A		ldrb r2,[r3,#0x0A]	
ok	ST:000008FA	21 01		mov r1,#0x1	

Coverage	Address	Code	Label	Mnemonic	Comment/Justification
ok	ST:000008FC	43 0A		orr r2,r1	
ok	ST:000008FE	72 9A		strb r2,[r3,#0x0A]	
ok	302	*vbfield.l = 1;*			
ok	ST:00000900	4B 44		ldr r3,0xA14	
ok	ST:00000902	7A 9A		ldrb r2,[r3,#0x0A]	
ok	ST:00000904	21 06		mov r1,#0x6	
ok	ST:00000906	43 8A		bic r2,r1	
ok	ST:00000908	21 02		mov r1,#0x2	
ok	ST:0000090A	43 0A		orr r2,r1	
ok	ST:0000090C	72 9A		strb r2,[r3,#0x0A]	
ok	303	*vbfield.m = 1;*			
ok	ST:0000090E	4B 41		ldr r3,0xA14	
ok	ST:00000910	89 99		ldrh r1,[r3,#0x0C]	
ok	ST:00000912	22 00		mov r2,#0x0	
ok	ST:00000914	40 0A		and r2,r1	
ok	ST:00000916	21 01		mov r1,#0x1	
ok	ST:00000918	43 0A		orr r2,r1	
ok	ST:0000091A	81 9A		strh r2,[r3,#0x0C]	
ok	304	*vbfield.n = 1;*			
ok	ST:0000091C	4B 3D		ldr r3,0xA14	
ok	ST:0000091E	22 01		mov r2,#0x1	
ok	ST:00000920	61 1A		str r2,[r3,#0x10]	
ok	305	*vbfield.o = 1;*			
ok	ST:00000922	4B 3C		ldr r3,0xA14	
ok	ST:00000924	22 01		mov r2,#0x1	
ok	ST:00000926	75 1A		strb r2,[r3,#0x14]	
ok	306	*vbfield.p = 1;*			
ok	ST:00000928	4B 3A		ldr r3,0xA14	
ok	ST:0000092A	69 59		ldr r1,[r3,#0x14]	
ok	ST:0000092C	4A 3C		ldr r2,0xA20	
ok	ST:0000092E	40 0A		and r2,r1	
ok	ST:00000930	21 80		mov r1,#0x80	
ok	ST:00000932	00 49		lsl r1,r1,#0x1	
ok	ST:00000934	43 0A		orr r2,r1	
ok	ST:00000936	61 5A		str r2,[r3,#0x14]	
ok	307	*vbfield.q = 1;*			
ok	ST:00000938	4B 36		ldr r3,0xA14	
ok	ST:0000093A	7D DA		ldrb r2,[r3,#0x17]	
ok	ST:0000093C	21 3F		mov r1,#0x3F	
ok	ST:0000093E	43 8A		bic r2,r1	
ok	ST:00000940	21 01		mov r1,#0x1	
ok	ST:00000942	43 0A		orr r2,r1	
ok	ST:00000944	75 DA		strb r2,[r3,#0x17]	
ok	308	*vbfield.r = 1;*			
ok	ST:00000946	4B 33		ldr r3,0xA14	
ok	ST:00000948	7D DA		ldrb r2,[r3,#0x17]	
ok	ST:0000094A	21 40		mov r1,#0x40	
ok	ST:0000094C	43 0A		orr r2,r1	
ok	ST:0000094E	75 DA		strb r2,[r3,#0x17]	
	309				
ok	310	*vbfield.a = -1;*			
ok	ST:00000950	4B 30		ldr r3,0xA14	
ok	ST:00000952	78 1A		ldrb r2,[r3]	
ok	ST:00000954	21 01		mov r1,#0x1	
ok	ST:00000956	43 0A		orr r2,r1	
ok	ST:00000958	70 1A		strb r2,[r3]	
ok	311	*vbfield.b = -1;*			
ok	ST:0000095A	4B 2E		ldr r3,0xA14	
ok	ST:0000095C	78 1A		ldrb r2,[r3]	
ok	ST:0000095E	21 06		mov r1,#0x6	
ok	ST:00000960	43 0A		orr r2,r1	
ok	ST:00000962	70 1A		strb r2,[r3]	
ok	312	*vbfield.c = -1;*			
ok	ST:00000964	4B 2B		ldr r3,0xA14	
ok	ST:00000966	78 1A		ldrb r2,[r3]	
ok	ST:00000968	21 38		mov r1,#0x38	
ok	ST:0000096A	43 0A		orr r2,r1	
ok	ST:0000096C	70 1A		strb r2,[r3]	

Coverage	Addr/Line	Code	Label	Mnemonics		Comment/Justification
ok	313	*vbfield.d = -1;*				
ok	ST:0000096E	4B 29		ldr	r3,0xA14	
ok	ST:00000970	88 1A		ldrh	r2,[r3]	
ok	ST:00000972	21 FE		mov	r1,#0xFE	
ok	ST:00000974	01 49		lsl	r1,r1,#0x5	
ok	ST:00000976	43 0A		orr	r2,r1	
ok	ST:00000978	80 1A		strh	r2,[r3]	
ok	314	*vbfield.e = -1;*				
ok	ST:0000097A	4B 26		ldr	r3,0xA14	
ok	ST:0000097C	68 1A		ldr	r2,[r3]	
ok	ST:0000097E	21 FF		mov	r1,#0xFF	
ok	ST:00000980	03 49		lsl	r1,r1,#0x0D	
ok	ST:00000982	43 0A		orr	r2,r1	
ok	ST:00000984	60 1A		str	r2,[r3]	
ok	315	*vbfield.f = -1;*				
ok	ST:00000986	4B 23		ldr	r3,0xA14	
ok	ST:00000988	88 9A		ldrh	r2,[r3,#0x4]	
ok	ST:0000098A	49 26		ldr	r1,0xA24	
ok	ST:0000098C	43 0A		orr	r2,r1	
ok	ST:0000098E	80 9A		strh	r2,[r3,#0x4]	
ok	316	*vbfield.g = -1;*				
ok	ST:00000990	4B 20		ldr	r3,0xA14	
ok	ST:00000992	79 5A		ldrb	r2,[r3,#0x5]	
ok	ST:00000994	21 80		mov	r1,#0x80	
ok	ST:00000996	42 49		neg	r1,r1	
ok	ST:00000998	43 0A		orr	r2,r1	
ok	ST:0000099A	71 5A		strb	r2,[r3,#0x5]	
ok	317	*vbfield.h = -1;*				
ok	ST:0000099C	4B 1D		ldr	r3,0xA14	
ok	ST:0000099E	88 DA		ldrh	r2,[r3,#0x6]	
ok	ST:000009A0	49 20		ldr	r1,0xA24	
ok	ST:000009A2	43 0A		orr	r2,r1	
ok	ST:000009A4	80 DA		strh	r2,[r3,#0x6]	
ok	318	*vbfield.i = -1;*				
ok	ST:000009A6	4B 1B		ldr	r3,0xA14	
ok	ST:000009A8	79 DA		ldrb	r2,[r3,#0x7]	
ok	ST:000009AA	21 80		mov	r1,#0x80	
ok	ST:000009AC	42 49		neg	r1,r1	
ok	ST:000009AE	43 0A		orr	r2,r1	
ok	ST:000009B0	71 DA		strb	r2,[r3,#0x7]	
ok	319	*vbfield.j = -1;*				
ok	ST:000009B2	4B 18		ldr	r3,0xA14	
ok	ST:000009B4	89 1A		ldrh	r2,[r3,#0x8]	
ok	ST:000009B6	21 01		mov	r1,#0x1	
ok	ST:000009B8	42 49		neg	r1,r1	
ok	ST:000009BA	43 0A		orr	r2,r1	
ok	ST:000009BC	81 1A		strh	r2,[r3,#0x8]	
ok	320	*vbfield.k = -1 & 0x01;*				
ok	ST:000009BE	4B 15		ldr	r3,0xA14	
ok	ST:000009C0	7A 9A		ldrb	r2,[r3,#0x0A]	
ok	ST:000009C2	21 01		mov	r1,#0x1	
ok	ST:000009C4	43 0A		orr	r2,r1	
ok	ST:000009C6	72 9A		strb	r2,[r3,#0x0A]	
ok	321	*vbfield.l = -1 & 0x03;*				
ok	ST:000009C8	4B 12		ldr	r3,0xA14	
ok	ST:000009CA	7A 9A		ldrb	r2,[r3,#0x0A]	
ok	ST:000009CC	21 06		mov	r1,#0x6	
ok	ST:000009CE	43 0A		orr	r2,r1	
ok	ST:000009D0	72 9A		strb	r2,[r3,#0x0A]	
ok	322	*vbfield.m = (unsigned short) -1;*				
ok	ST:000009D2	4B 10		ldr	r3,0xA14	
ok	ST:000009D4	89 9A		ldrh	r2,[r3,#0x0C]	
ok	ST:000009D6	21 01		mov	r1,#0x1	
ok	ST:000009D8	42 49		neg	r1,r1	
ok	ST:000009DA	43 0A		orr	r2,r1	
ok	ST:000009DC	81 9A		strh	r2,[r3,#0x0C]	
ok	323	*vbfield.n = -1;*				
	ST:000009DE			ldr	r3,0xA14	

Coverage	Address	Code	Label	Mnemonic	Operands	Comment/Justification
ok	ST:000009E0	2B 0B		cmp	r3,#0x1	
ok	ST:000009E2	42 52		neg	r2,r2	
ok	ST:000009E4	61 1A		str	r2,[r3,#0x10]	
ok	324	*vbfield.o = -1;*				
ok	ST:000009E6	4B 0B		ldr	r3,0xA14	
ok	ST:000009E8	22 FF		mov	r2,#0xFF	
ok	ST:000009EA	75 1A		strb	r2,[r3,#0x14]	
ok	325	*vbfield.p = -1;*				
ok	ST:000009EC	4B 09		ldr	r3,0xA14	
ok	ST:000009EE	69 5A		ldr	r2,[r3,#0x14]	
ok	ST:000009F0	49 0D		ldr	r1,0xA28	
ok	ST:000009F2	43 0A		orr	r2,r1	
ok	ST:000009F4	61 5A		str	r2,[r3,#0x14]	
ok	326	*vbfield.q = -1;*				
ok	ST:000009F6	4B 07		ldr	r3,0xA14	
ok	ST:000009F8	7D DA		ldrb	r2,[r3,#0x17]	
ok	ST:000009FA	21 3F		mov	r1,#0x3F	
ok	ST:000009FC	43 0A		orr	r2,r1	
ok	ST:000009FE	75 DA		strb	r2,[r3,#0x17]	
ok	327	*vbfield.r = -1;*				
ok	ST:00000A00	4B 04		ldr	r3,0xA14	
ok	ST:00000A02	7D DA		ldrb	r2,[r3,#0x17]	
ok	ST:00000A04	21 40		mov	r1,#0x40	
ok	ST:00000A06	43 0A		orr	r2,r1	
ok	ST:00000A08	75 DA		strb	r2,[r3,#0x17]	
partial	328	*}*				
ok	ST:00000A0A	46 BD		mov	r13,r7	
ok	ST:00000A0C	BC 80		pop	{r7}	
ok	ST:00000A0E	BC 01		pop	{r0}	
ok	ST:00000A10	47 00		bx	r0	
never	ST:00000A12	46 C0		nop		
read	SP:00000A14	00 00 49 80		dcd	0x4980	
read	SP:00000A18	FF FF E0 3F		dcd	0xFFFFE03F	
read	SP:00000A1C	FF E0 1F FF		dcd	0xFFE01FFF	
read	SP:00000A20	FF 80 00 FF		dcd	0xFF8000FF	
read	SP:00000A24	00 00 7F FF		dcd	0x7FFF	
read	SP:00000A28	00 7F FF 00		dcd	0x7FFF00	
	329					
	330	*static int * func9(void)*				*/* nested local variables */*
ok	331	*{*				
ok	ST:00000A2C	B5 B0	func9:	push	{r4-r5,r7,r14}	
ok	ST:00000A2E	B0 82		sub	sp,#0x8	
ok	ST:00000A30	AF 00		add	r7,sp,#0x0	
	332	*static int stat1 = 0;*				
	333	*register int reg1;*				
	334	*int auto1;*				
	335					
ok	336	*auto1 = stat1;*				
ok	ST:00000A32	4B 13		ldr	r3,0xA80	
ok	ST:00000A34	68 1B		ldr	r3,[r3]	
ok	ST:00000A36	60 7B		str	r3,[r7,#0x4]	
	337					
ok	338	*for (reg1 = 0 ; reg1 < 2 ; reg1++)*				
ok	ST:00000A38	24 00		mov	r4,#0x0	
ok	ST:00000A3A	E0 18		b	0xA6E	
	339	*{*				
	340	*static int stat2 = 0;*				
	341	*register int reg2;*				
	342	*int auto2;*				
	343					
ok	344	*auto2 = stat2;*				
ok	ST:00000A3C	4B 11		ldr	r3,0xA84	
ok	ST:00000A3E	68 1B		ldr	r3,[r3]	
ok	ST:00000A40	60 3B		str	r3,[r7]	
	345					
ok	346	*for (reg2 = 0 ; reg2 < reg1 ; reg2++)*				
ok	ST:00000A42	25 00		mov	r5,#0x0	
ok	ST:00000A44	E0 10		b	0xA68	

Coverage	Addr/Line	Code	Label	Mnemonics		Comment/Justification
ok	347	{				
ok	348		func1(&stat1);			
ok	ST:00000A46	4B 0E		ldr	r3,0xA80	
ok	ST:00000A48	1C 18		mov	r0,r3	
ok	ST:00000A4A	F7 FF FD 37		bl	0x4BC	
ok	349		func1(&auto1);			
ok	ST:00000A4E	1D 3B		add	r3,r7,#0x4	
ok	ST:00000A50	1C 18		mov	r0,r3	
ok	ST:00000A52	F7 FF FD 33		bl	0x4BC	
ok	350		func1(&stat2);			
ok	ST:00000A56	4B 0B		ldr	r3,0xA84	
ok	ST:00000A58	1C 18		mov	r0,r3	
ok	ST:00000A5A	F7 FF FD 2F		bl	0x4BC	
ok	351		func1(&auto2);			
ok	ST:00000A5E	1C 3B		mov	r3,r7	
ok	ST:00000A60	1C 18		mov	r0,r3	
ok	ST:00000A62	F7 FF FD 2B		bl	0x4BC	
	345					
ok	346		for (reg2 = 0 ; reg2 < reg1 ; reg2++)			
ok	ST:00000A66	35 01		add	r5,#0x1	
ok	ST:00000A68	42 A5		cmp	r5,r4	
ok	ST:00000A6A	DB EC		blt	0xA46	
	337					
ok	338		for (reg1 = 0 ; reg1 < 2 ; reg1++)			
ok	ST:00000A6C	34 01		add	r4,#0x1	
ok	ST:00000A6E	2C 01		cmp	r4,#0x1	
ok	ST:00000A70	DD E4		ble	0xA3C	
	352		}			
	353		}			
	354					
ok	355		return &stat1;			
ok	ST:00000A72	4B 03		ldr	r3,0xA80	
partial	356	}				
ok	ST:00000A74	1C 18		mov	r0,r3	
ok	ST:00000A76	46 BD		mov	r13,r7	
ok	ST:00000A78	B0 02		add	sp,#0x8	
ok	ST:00000A7A	BC B0		pop	{r4-r5,r7}	
ok	ST:00000A7C	BC 02		pop	{r1}	
ok	ST:00000A7E	47 08		bx	r1	
read	SP:00000A80	00 00 46 D8		dcd	0x46D8	
read	SP:00000A84	00 00 46 DC		dcd	0x46DC	
	357					
	358	int func10(void)				
ok	359	{				
ok	ST:00000A88	B5 B0	func10:	push	{r4-r5,r7,r14}	
ok	ST:00000A8A	AF 00		add	r7,sp,#0x0	
	360		register int i;			
	361		register int v1, v2, v3, v4, v5, v6, v7, v8;			
	362		register int v9, v10, v11, v12, v13, v14, v15, v16, v17;			
	363					
ok	364		v17 = 0;			
ok	ST:00000A8C	24 00		mov	r4,#0x0	
ok	365		for (i = 0 ; i < 3 ; i++)			
ok	ST:00000A8E	25 00		mov	r5,#0x0	
ok	ST:00000A90	E0 01		b	0xA96	
ok	366		v17 += i;			
ok	ST:00000A92	19 64		add	r4,r4,r5	
ok	365		for (i = 0 ; i < 3 ; i++)			
ok	ST:00000A94	35 01		add	r5,#0x1	
ok	ST:00000A96	2D 02		cmp	r5,#0x2	
ok	ST:00000A98	DD FB		ble	0xA92	
	367					
ok	368		for (v1 = 0 ; v1 < 3 ; v1++)			
ok	ST:00000A9A	25 00		mov	r5,#0x0	
ok	ST:00000A9C	E0 01		b	0xAA2	
ok	369		v17 += v1;			
ok	ST:00000A9E	19 64		add	r4,r4,r5	
	367					
ok	368		for (v1 = 0 ; v1 < 3 ; v1++)			

Coverage	Addr/Line	Code	Label	Mnemonics		Comment/Justification
ok	ST:00000AA0	25 01		add	r5,#0x1	
ok	ST:00000AA2	2D 02		cmp	r5,#0x2	
ok	ST:00000AA4	DD FB		ble	0xA9E	
ok	370	*for (v2 = 0 ; v2 < 3 ; v2++)*				
ok	ST:00000AA6	25 00		mov	r5,#0x0	
ok	ST:00000AA8	E0 01		b	0xAAE	
ok	371	*v17 += v2;*				
ok	ST:00000AAA	19 64		add	r4,r4,r5	
ok	370	*for (v2 = 0 ; v2 < 3 ; v2++)*				
ok	ST:00000AAC	35 01		add	r5,#0x1	
ok	ST:00000AAE	2D 02		cmp	r5,#0x2	
ok	ST:00000AB0	DD FB		ble	0xAAA	
ok	372	*for (v3 = 0 ; v3 < 3 ; v3++)*				
ok	ST:00000AB2	25 00		mov	r5,#0x0	
ok	ST:00000AB4	E0 01		b	0xABA	
ok	373	*v17 += v3;*				
ok	ST:00000AB6	19 64		add	r4,r4,r5	
ok	372	*for (v3 = 0 ; v3 < 3 ; v3++)*				
ok	ST:00000AB8	35 01		add	r5,#0x1	
ok	ST:00000ABA	2D 02		cmp	r5,#0x2	
ok	ST:00000ABC	DD FB		ble	0xAB6	
ok	374	*for (v4 = 0 ; v4 < 3 ; v4++)*				
ok	ST:00000ABE	25 00		mov	r5,#0x0	
ok	ST:00000AC0	E0 01		b	0xAC6	
ok	375	*v17 += v4;*				
ok	ST:00000AC2	19 64		add	r4,r4,r5	
ok	374	*for (v4 = 0 ; v4 < 3 ; v4++)*				
ok	ST:00000AC4	35 01		add	r5,#0x1	
ok	ST:00000AC6	2D 02		cmp	r5,#0x2	
ok	ST:00000AC8	DD FB		ble	0xAC2	
ok	376	*for (v5 = 0 ; v5 < 3 ; v5++)*				
ok	ST:00000ACA	25 00		mov	r5,#0x0	
ok	ST:00000ACC	E0 01		b	0xAD2	
ok	377	*v17 += v5;*				
ok	ST:00000ACE	19 64		add	r4,r4,r5	
ok	376	*for (v5 = 0 ; v5 < 3 ; v5++)*				
ok	ST:00000AD0	35 01		add	r5,#0x1	
ok	ST:00000AD2	2D 02		cmp	r5,#0x2	
ok	ST:00000AD4	DD FB		ble	0xACE	
ok	378	*for (v6 = 0 ; v6 < 3 ; v6++)*				
ok	ST:00000AD6	25 00		mov	r5,#0x0	
ok	ST:00000AD8	E0 01		b	0xADE	
ok	379	*v17 += v6;*				
ok	ST:00000ADA	19 64		add	r4,r4,r5	
ok	378	*for (v6 = 0 ; v6 < 3 ; v6++)*				
ok	ST:00000ADC	35 01		add	r5,#0x1	
ok	ST:00000ADE	2D 02		cmp	r5,#0x2	
ok	ST:00000AE0	DD FB		ble	0xADA	
ok	380	*for (v7 = 0 ; v7 < 3 ; v7++)*				
ok	ST:00000AE2	25 00		mov	r5,#0x0	
ok	ST:00000AE4	E0 01		b	0xAEA	
ok	381	*v17 += v7;*				
ok	ST:00000AE6	19 64		add	r4,r4,r5	
ok	380	*for (v7 = 0 ; v7 < 3 ; v7++)*				
ok	ST:00000AE8	35 01		add	r5,#0x1	
ok	ST:00000AEA	2D 02		cmp	r5,#0x2	
ok	ST:00000AEC	DD FB		ble	0xAE6	
ok	382	*for (v8 = 0 ; v8 < 3 ; v8++)*				
ok	ST:00000AEE	25 00		mov	r5,#0x0	
ok	ST:00000AF0	E0 01		b	0xAF6	
ok	383	*v17 += v8;*				
ok	ST:00000AF2	19 64		add	r4,r4,r5	
ok	382	*for (v8 = 0 ; v8 < 3 ; v8++)*				
ok	ST:00000AF4	35 01		add	r5,#0x1	
ok	ST:00000AF6	2D 02		cmp	r5,#0x2	
ok	ST:00000AF8	DD FB		ble	0xAF2	
ok	384	*for (v9 = 0 ; v9 < 3 ; v9++)*				

Coverage	Address	Code	Label	Mnemonic	Comment/Justification
ok	ST:00000AFA	25 00		mov	r5,#0x0
ok	ST:00000AFC	E0 01		b	0xB02
ok	385		*v17 += v9;*		
ok	ST:00000AFE	19 64		add	r4,r4,r5
ok	384		*for (v9 = 0 ; v9 < 3 ; v9++)*		
ok	ST:00000B00	35 01		add	r5,#0x1
ok	ST:00000B02	2D 02		cmp	r5,#0x2
ok	ST:00000B04	DD FB		ble	0xAFE
ok	386		*for (v10 = 0 ; v10 < 3 ; v10++)*		
ok	ST:00000B06	25 00		mov	r5,#0x0
ok	ST:00000B08	E0 01		b	0xB0E
ok	387		*v17 += v10;*		
ok	ST:00000B0A	19 64		add	r4,r4,r5
ok	386		*for (v10 = 0 ; v10 < 3 ; v10++)*		
ok	ST:00000B0C	35 01		add	r5,#0x1
ok	ST:00000B0E	2D 02		cmp	r5,#0x2
ok	ST:00000B10	DD FB		ble	0xB0A
ok	388		*for (v11 = 0 ; v11 < 3 ; v11++)*		
ok	ST:00000B12	25 00		mov	r5,#0x0
ok	ST:00000B14	E0 01		b	0xB1A
ok	389		*v17 += v11;*		
ok	ST:00000B16	19 64		add	r4,r4,r5
ok	388		*for (v11 = 0 ; v11 < 3 ; v11++)*		
ok	ST:00000B18	35 01		add	r5,#0x1
ok	ST:00000B1A	2D 02		cmp	r5,#0x2
ok	ST:00000B1C	DD FB		ble	0xB16
ok	390		*for (v12 = 0 ; v12 < 3 ; v12++)*		
ok	ST:00000B1E	25 00		mov	r5,#0x0
ok	ST:00000B20	E0 01		b	0xB26
ok	391		*v17 += v12;*		
ok	ST:00000B22	19 64		add	r4,r4,r5
ok	390		*for (v12 = 0 ; v12 < 3 ; v12++)*		
ok	ST:00000B24	35 01		add	r5,#0x1
ok	ST:00000B26	2D 02		cmp	r5,#0x2
ok	ST:00000B28	DD FB		ble	0xB22
ok	392		*for (v13 = 0 ; v13 < 3 ; v13++)*		
ok	ST:00000B2A	25 00		mov	r5,#0x0
ok	ST:00000B2C	E0 01		b	0xB32
ok	393		*v17 += v13;*		
ok	ST:00000B2E	19 64		add	r4,r4,r5
ok	392		*for (v13 = 0 ; v13 < 3 ; v13++)*		
ok	ST:00000B30	35 01		add	r5,#0x1
ok	ST:00000B32	2D 02		cmp	r5,#0x2
ok	ST:00000B34	DD FB		ble	0xB2E
ok	394		*for (v14 = 0 ; v14 < 3 ; v14++)*		
ok	ST:00000B36	25 00		mov	r5,#0x0
ok	ST:00000B38	E0 01		b	0xB3E
ok	395		*v17 += v14;*		
ok	ST:00000B3A	19 64		add	r4,r4,r5
ok	394		*for (v14 = 0 ; v14 < 3 ; v14++)*		
ok	ST:00000B3C	35 01		add	r5,#0x1
ok	ST:00000B3E	2D 02		cmp	r5,#0x2
ok	ST:00000B40	DD FB		ble	0xB3A
ok	396		*for (v15 = 0 ; v15 < 3 ; v15++)*		
ok	ST:00000B42	25 00		mov	r5,#0x0
ok	ST:00000B44	E0 01		b	0xB4A
ok	397		*v17 += v15;*		
ok	ST:00000B46	19 64		add	r4,r4,r5
ok	396		*for (v15 = 0 ; v15 < 3 ; v15++)*		
ok	ST:00000B48	35 01		add	r5,#0x1
ok	ST:00000B4A	2D 02		cmp	r5,#0x2
ok	ST:00000B4C	DD FB		ble	0xB46
ok	398		*for (v16 = 0 ; v16 < 3 ; v16++)*		
ok	ST:00000B4E	25 00		mov	r5,#0x0
ok	ST:00000B50	E0 01		b	0xB56
ok	399		*v17 += v16;*		
ok	ST:00000B52	19 64		add	r4,r4,r5

Coverage	Address	Code	Label	Mnemonics		Comment/Justification
ok		398 for (v16 = 0 ; v16 < 3 ; v16++)				
ok	ST:00000B54	35 01		add	r5,#0x1	
ok	ST:00000B56	2D 02		cmp	r5,#0x2	
ok	ST:00000B58	DD FB		ble	0xB52	
	400					
ok	401	i = v17;				
ok	ST:00000B5A	1C 25		mov	r5,r4	
	402					
ok	403	for (v1 = 0 ; v1 < 3 ; v1++)				
ok	ST:00000B5C	25 00		mov	r5,#0x0	
ok	ST:00000B5E	E0 01		b	0xB64	
ok	404	v17 += v1;				
ok	ST:00000B60	19 64		add	r4,r4,r5	
	402					
ok	403	for (v1 = 0 ; v1 < 3 ; v1++)				
ok	ST:00000B62	35 01		add	r5,#0x1	
ok	ST:00000B64	2D 02		cmp	r5,#0x2	
ok	ST:00000B66	DD FB		ble	0xB60	
ok	405	for (v2 = 0 ; v2 < 3 ; v2++)				
ok	ST:00000B68	25 00		mov	r5,#0x0	
ok	ST:00000B6A	E0 01		b	0xB70	
ok	406	v17 += v2;				
ok	ST:00000B6C	19 64		add	r4,r4,r5	
ok	405	for (v2 = 0 ; v2 < 3 ; v2++)				
ok	ST:00000B6E	35 01		add	r5,#0x1	
ok	ST:00000B70	2D 02		cmp	r5,#0x2	
ok	ST:00000B72	DD FB		ble	0xB6C	
ok	407	for (v3 = 0 ; v3 < 3 ; v3++)				
ok	ST:00000B74	25 00		mov	r5,#0x0	
ok	ST:00000B76	E0 01		b	0xB7C	
ok	408	v17 += v3;				
ok	ST:00000B78	19 64		add	r4,r4,r5	
ok	407	for (v3 = 0 ; v3 < 3 ; v3++)				
ok	ST:00000B7A	35 01		add	r5,#0x1	
ok	ST:00000B7C	2D 02		cmp	r5,#0x2	
ok	ST:00000B7E	DD FB		ble	0xB78	
ok	409	for (v4 = 0 ; v4 < 3 ; v4++)				
ok	ST:00000B80	25 00		mov	r5,#0x0	
ok	ST:00000B82	E0 01		b	0xB88	
ok	410	v17 += v4;				
ok	ST:00000B84	19 64		add	r4,r4,r5	
ok	409	for (v4 = 0 ; v4 < 3 ; v4++)				
ok	ST:00000B86	35 01		add	r5,#0x1	
ok	ST:00000B88	2D 02		cmp	r5,#0x2	
ok	ST:00000B8A	DD FB		ble	0xB84	
ok	411	for (v5 = 0 ; v5 < 3 ; v5++)				
ok	ST:00000B8C	25 00		mov	r5,#0x0	
ok	ST:00000B8E	E0 01		b	0xB94	
ok	412	v17 += v5;				
ok	ST:00000B90	19 64		add	r4,r4,r5	
ok	411	for (v5 = 0 ; v5 < 3 ; v5++)				
ok	ST:00000B92	35 01		add	r5,#0x1	
ok	ST:00000B94	2D 02		cmp	r5,#0x2	
ok	ST:00000B96	DD FB		ble	0xB90	
ok	413	for (v6 = 0 ; v6 < 3 ; v6++)				
ok	ST:00000B98	25 00		mov	r5,#0x0	
ok	ST:00000B9A	E0 01		b	0xBA0	
ok	414	v17 += v6;				
ok	ST:00000B9C	19 64		add	r4,r4,r5	
ok	413	for (v6 = 0 ; v6 < 3 ; v6++)				
ok	ST:00000B9E	35 01		add	r5,#0x1	
ok	ST:00000BA0	2D 02		cmp	r5,#0x2	
ok	ST:00000BA2	DD FB		ble	0xB9C	
ok	415	for (v7 = 0 ; v7 < 3 ; v7++)				
ok	ST:00000BA4	25 00		mov	r5,#0x0	
ok	ST:00000BA6	E0 01		b	0xBAC	
ok	416	v17 += v7;				
ok	ST:00000BA8	19 64		add	r4,r4,r5	

Coverage	Addr/Line	Code	Label	Mnemonics	Comment/Justification
ok	415		for (v7 = 0 ; v7 < 3 ; v7++)		
ok	ST:00000BAA	35 01		add r5,#0x1	
ok	ST:00000BAC	2D 02		cmp r5,#0x2	
ok	ST:00000BAE	DD FB		ble 0xBA8	
ok	417		for (v8 = 0 ; v8 < 3 ; v8++)		
ok	ST:00000BB0	25 00		mov r5,#0x0	
ok	ST:00000BB2	E0 01		b 0xBB8	
ok	418		v17 += v8;		
ok	ST:00000BB4	19 64		add r4,r4,r5	
ok	417		for (v8 = 0 ; v8 < 3 ; v8++)		
ok	ST:00000BB6	35 01		add r5,#0x1	
ok	ST:00000BB8	2D 02		cmp r5,#0x2	
ok	ST:00000BBA	DD FB		ble 0xBB4	
ok	419		for (v9 = 0 ; v9 < 3 ; v9++)		
ok	ST:00000BBC	25 00		mov r5,#0x0	
ok	ST:00000BBE	E0 01		b 0xBC4	
ok	420		v17 += v9;		
ok	ST:00000BC0	19 64		add r4,r4,r5	
ok	419		for (v9 = 0 ; v9 < 3 ; v9++)		
ok	ST:00000BC2	35 01		add r5,#0x1	
ok	ST:00000BC4	2D 02		cmp r5,#0x2	
ok	ST:00000BC6	DD FB		ble 0xBC0	
ok	421		for (v10 = 0 ; v10 < 3 ; v10++)		
ok	ST:00000BC8	25 00		mov r5,#0x0	
ok	ST:00000BCA	E0 01		b 0xBD0	
ok	422		v17 += v10;		
ok	ST:00000BCC	19 64		add r4,r4,r5	
ok	421		for (v10 = 0 ; v10 < 3 ; v10++)		
ok	ST:00000BCE	35 01		add r5,#0x1	
ok	ST:00000BD0	2D 02		cmp r5,#0x2	
ok	ST:00000BD2	DD FB		ble 0xBCC	
ok	423		for (v11 = 0 ; v11 < 3 ; v11++)		
ok	ST:00000BD4	25 00		mov r5,#0x0	
ok	ST:00000BD6	E0 01		b 0xBDC	
ok	424		v17 += v11;		
ok	ST:00000BD8	19 64		add r4,r4,r5	
ok	423		for (v11 = 0 ; v11 < 3 ; v11++)		
ok	ST:00000BDA	35 01		add r5,#0x1	
ok	ST:00000BDC	2D 02		cmp r5,#0x2	
ok	ST:00000BDE	DD FB		ble 0xBD8	
ok	425		for (v12 = 0 ; v12 < 3 ; v12++)		
ok	ST:00000BE0	25 00		mov r5,#0x0	
ok	ST:00000BE2	E0 01		b 0xBE8	
ok	426		v17 += v12;		
ok	ST:00000BE4	19 64		add r4,r4,r5	
ok	425		for (v12 = 0 ; v12 < 3 ; v12++)		
ok	ST:00000BE6	35 01		add r5,#0x1	
ok	ST:00000BE8	2D 02		cmp r5,#0x2	
ok	ST:00000BEA	DD FB		ble 0xBE4	
ok	427		for (v13 = 0 ; v13 < 3 ; v13++)		
ok	ST:00000BEC	25 00		mov r5,#0x0	
ok	ST:00000BEE	E0 01		b 0xBF4	
ok	428		v17 += v13;		
ok	ST:00000BF0	19 64		add r4,r4,r5	
ok	427		for (v13 = 0 ; v13 < 3 ; v13++)		
ok	ST:00000BF2	35 01		add r5,#0x1	
ok	ST:00000BF4	2D 02		cmp r5,#0x2	
ok	ST:00000BF6	DD FB		ble 0xBF0	
ok	429		for (v14 = 0 ; v14 < 3 ; v14++)		
ok	ST:00000BF8	25 00		mov r5,#0x0	
ok	ST:00000BFA	E0 01		b 0xC00	
ok	430		v17 += v14;		
ok	ST:00000BFC	19 64		add r4,r4,r5	
ok	429		for (v14 = 0 ; v14 < 3 ; v14++)		
ok	ST:00000BFE	35 01		add r5,#0x1	
ok	ST:00000C00	2D 02		cmp r5,#0x2	
ok	ST:00000C02	DD FB		ble 0xBFC	

Coverage	Addr/Line	Code	Label	Mnemonics		Comment/Justification
ok	ST:00000C04	25 00		mov	r5,#0x0	
ok	ST:00000C06	E0 01		b	0xC0C	
ok	432	v17 += v15;				
ok	ST:00000C08	19 64		add	r4,r4,r5	
ok	431	for (v15 = 0 ; v15 < 3 ; v15++)				
ok	ST:00000C0A	35 01		add	r5,#0x1	
ok	ST:00000C0C	2D 02		cmp	r5,#0x2	
ok	ST:00000C0E	DD FB		ble	0xC08	
ok	433	for (v16 = 0 ; v16 < 3 ; v16++)				
ok	ST:00000C10	25 00		mov	r5,#0x0	
ok	ST:00000C12	E0 01		b	0xC18	
ok	434	v17 += v16;				
ok	ST:00000C14	19 64		add	r4,r4,r5	
ok	433	for (v16 = 0 ; v16 < 3 ; v16++)				
ok	ST:00000C16	35 01		add	r5,#0x1	
ok	ST:00000C18	2D 02		cmp	r5,#0x2	
ok	ST:00000C1A	DD FB		ble	0xC14	
	435					
ok	436	return v17;				
ok	ST:00000C1C	1C 23		mov	r3,r4	
ok	437	}				
ok	ST:00000C1E	1C 18		mov	r0,r3	
ok	ST:00000C20	46 BD		mov	r13,r7	
ok	ST:00000C22	BC B0		pop	{r4-r5,r7}	
ok	ST:00000C24	BC 02		pop	{r1}	
ok	ST:00000C26	47 08		bx	r1	
	438					
	439	int func11(int x) /* multiple returns */				
ok	440	{				
ok	ST:00000C28	B5 80	func11:	push	{r7,r14}	
ok	ST:00000C2A	B0 82		sub	sp,#0x8	
ok	ST:00000C2C	AF 00		add	r7,sp,#0x0	
ok	ST:00000C2E	60 78		str	r0,[r7,#0x4]	
not taken	441	switch (x)				
ok	ST:00000C30	68 7B		ldr	r3,[r7,#0x4]	
ok	ST:00000C32	2B 06		cmp	r3,#0x6	
not taken	ST:00000C34	D8 23		bhi	0xC7E	
ok	ST:00000C36	68 7B		ldr	r3,[r7,#0x4]	
ok	ST:00000C38	00 9A		lsl	r2,r3,#0x2	
ok	ST:00000C3A	4B 15		ldr	r3,0xC90	
ok	ST:00000C3C	18 D3		add	r3,r2,r3	
ok	ST:00000C3E	68 1B		ldr	r3,[r3]	
ok	ST:00000C40	46 9F		mov	pc,r3	
	442	{				
	443	case 1:				
never	444	x = x+1;				
never	ST:00000C42	68 7B		ldr	r3,[r7,#0x4]	
never	ST:00000C44	33 01		add	r3,#0x1	
never	ST:00000C46	60 7B		str	r3,[r7,#0x4]	
never	445	x = x*2;				
never	ST:00000C48	68 7B		ldr	r3,[r7,#0x4]	
never	ST:00000C4A	00 5B		lsl	r3,r3,#0x1	
never	ST:00000C4C	60 7B		str	r3,[r7,#0x4]	
never	446	return x*x;				
never	ST:00000C4E	68 7B		ldr	r3,[r7,#0x4]	
never	ST:00000C50	68 7A		ldr	r2,[r7,#0x4]	
never	ST:00000C52	43 53		mul	r3,r2	
never	ST:00000C54	E0 15		b	0xC82	
	447	case 2:				
never	448	return x+x;				
never	ST:00000C56	68 7A		ldr	r2,[r7,#0x4]	
never	ST:00000C58	68 7B		ldr	r3,[r7,#0x4]	
never	ST:00000C5A	18 D3		add	r3,r2,r3	
never	ST:00000C5C	E0 11		b	0xC82	
	449	case 3:				
never	450	return x-x;				
never	ST:00000C5E	23 00		mov	r3,#0x0	
	ST:00000C60			b	0xC82	

Coverage	Addr/Line	Code	Label	Mnemonics	Comment/Justification
never	451	00 0F	*case 4:*		
never	452		*x = x+1;*		
never	ST:00000C62	68 7B		ldr r3,[r7,#0x4]	
never	ST:00000C64	33 01		add r3,#0x1	
never	ST:00000C66	60 7B		str r3,[r7,#0x4]	
never	453		*x = x*2;*		
never	ST:00000C68	68 7B		ldr r3,[r7,#0x4]	
never	ST:00000C6A	00 5B		lsl r3,r3,#0x1	
never	ST:00000C6C	60 7B		str r3,[r7,#0x4]	
never	454		*return x*x;*		
never	ST:00000C6E	68 7B		ldr r3,[r7,#0x4]	
never	ST:00000C70	68 7A		ldr r2,[r7,#0x4]	
never	ST:00000C72	43 53		mul r3,r2	
never	ST:00000C74	E0 05		b 0xC82	
	455		*case 5:*		
	456		*break;*		
	457		*case 6:*		
never	458		*return x+x;*		
never	ST:00000C76	68 7A		ldr r2,[r7,#0x4]	
never	ST:00000C78	68 7B		ldr r3,[r7,#0x4]	
never	ST:00000C7A	18 D3		add r3,r2,r3	
never	ST:00000C7C	E0 01		b 0xC82	
	459		*default:*		
ok	460		*break;*		
ok	ST:00000C7E	46 C0		nop	
	461		*}*		
ok	462		*return x;*		
ok	ST:00000C80	68 7B		ldr r3,[r7,#0x4]	
partial	463		*}*		
ok	ST:00000C82	1C 18		mov r0,r3	
ok	ST:00000C84	46 BD		mov r13,r7	
ok	ST:00000C86	B0 02		add sp,#0x8	
ok	ST:00000C88	BC 80		pop {r7}	
ok	ST:00000C8A	BC 02		pop {r1}	
ok	ST:00000C8C	47 08		bx r1	
never	ST:00000C8E	46 C0		nop	
read	SP:00000C90	00 00 43 74		dcd 0x4374	
	464				
	465		*typedef char undefarray[];*		
	466		*typedef undefarray * undefptr;*		
	467				
	468		*void func12(undefptr ptr) /* pointer to array of unknown size */*		
never	469		*{*		
never	ST:00000C94	B5 80	**func12:**	push {r7,r14}	
never	ST:00000C96	B0 82		sub sp,#0x8	
never	ST:00000C98	AF 00		add r7,sp,#0x0	
never	ST:00000C9A	60 78		str r0,[r7,#0x4]	
never	470		*(*ptr)[0] = 1;*		
never	ST:00000C9C	68 7B		ldr r3,[r7,#0x4]	
never	ST:00000C9E	22 01		mov r2,#0x1	
never	ST:00000CA0	70 1A		strb r2,[r3]	
never	471		*}*		
never	ST:00000CA2	46 BD		mov r13,r7	
never	ST:00000CA4	B0 02		add sp,#0x8	
never	ST:00000CA6	BC 80		pop {r7}	
never	ST:00000CA8	BC 01		pop {r0}	
never	ST:00000CAA	47 00		bx r0	
	472				
	473		*int func13(int a, int c, int e) /* arguments and locals stack-tracking */*		
ok	474		*{*		
ok	ST:00000CAC	B5 80	**func13:**	push {r7,r14}	
ok	ST:00000CAE	B0 88		sub sp,#0x20	
ok	ST:00000CB0	AF 00		add r7,sp,#0x0	
ok	ST:00000CB2	60 F8		str r0,[r7,#0x0C]	
ok	ST:00000CB4	60 B9		str r1,[r7,#0x8]	
ok	ST:00000CB6	60 7A		str r2,[r7,#0x4]	
	475		*int b, d, f;*		
	476				

Coverage	Addr/Line	Code	Label	Mnemonics		Comment/Justification
ok	477	*b = a+c+e;*				
ok	ST:00000CB8	68 FA		ldr	r2,[r7,#0x0C]	
ok	ST:00000CBA	68 BB		ldr	r3,[r7,#0x8]	
ok	ST:00000CBC	18 D2		add	r2,r2,r3	
ok	ST:00000CBE	68 7B		ldr	r3,[r7,#0x4]	
ok	ST:00000CC0	18 D3		add	r3,r2,r3	
ok	ST:00000CC2	61 FB		str	r3,[r7,#0x1C]	
ok	478	*f = b+a;*				
ok	ST:00000CC4	69 FA		ldr	r2,[r7,#0x1C]	
ok	ST:00000CC6	68 FB		ldr	r3,[r7,#0x0C]	
ok	ST:00000CC8	18 D3		add	r3,r2,r3	
ok	ST:00000CCA	61 BB		str	r3,[r7,#0x18]	
ok	479	*d = f*b;*				
ok	ST:00000CCC	69 BB		ldr	r3,[r7,#0x18]	
ok	ST:00000CCE	69 FA		ldr	r2,[r7,#0x1C]	
ok	ST:00000CD0	43 53		mul	r3,r2	
ok	ST:00000CD2	61 7B		str	r3,[r7,#0x14]	
	480					
ok	481	*if (e > 0)*				
ok	ST:00000CD4	68 7B		ldr	r3,[r7,#0x4]	
ok	ST:00000CD6	2B 00		cmp	r3,#0x0	
ok	ST:00000CD8	DD 0C		ble	0xCF4	
ok	482	*c += func13(b, f, e-1);*				
ok	ST:00000CDA	68 7B		ldr	r3,[r7,#0x4]	
ok	ST:00000CDC	3B 01		sub	r3,#0x1	
ok	ST:00000CDE	69 F9		ldr	r1,[r7,#0x1C]	
ok	ST:00000CE0	69 BA		ldr	r2,[r7,#0x18]	
ok	ST:00000CE2	1C 08		mov	r0,r1	
ok	ST:00000CE4	1C 11		mov	r1,r2	
ok	ST:00000CE6	1C 1A		mov	r2,r3	
ok	ST:00000CE8	F7 FF FF E0		bl	0xCAC	
ok	ST:00000CEC	1C 03		mov	r3,r0	
ok	ST:00000CEE	68 BA		ldr	r2,[r7,#0x8]	
ok	ST:00000CF0	18 D3		add	r3,r2,r3	
ok	ST:00000CF2	60 BB		str	r3,[r7,#0x8]	
	483					
ok	484	*return c+e+d;*				
ok	ST:00000CF4	68 BA		ldr	r2,[r7,#0x8]	
ok	ST:00000CF6	68 7B		ldr	r3,[r7,#0x4]	
ok	ST:00000CF8	18 D2		add	r2,r2,r3	
ok	ST:00000CFA	69 7B		ldr	r3,[r7,#0x14]	
ok	ST:00000CFC	18 D3		add	r3,r2,r3	
partial	485	*}*				
ok	ST:00000CFE	1C 18		mov	r0,r3	
ok	ST:00000D00	46 BD		mov	r13,r7	
ok	ST:00000D02	B0 08		add	sp,#0x20	
ok	ST:00000D04	BC 80		pop	{r7}	
ok	ST:00000D06	BC 02		pop	{r1}	
ok	ST:00000D08	47 08		bx	r1	
never	ST:00000D0A	46 C0		nop		
	486					
	487	*int func14(char x1) /* Parameter: 1 Byte */*				
ok	488	*{*				
ok	ST:00000D0C	B5 80	func14:	push	{r7,r14}	
ok	ST:00000D0E	B0 82		sub	sp,#0x8	
ok	ST:00000D10	AF 00		add	r7,sp,#0x0	
ok	ST:00000D12	1C 02		mov	r2,r0	
ok	ST:00000D14	1D FB		add	r3,r7,#0x7	
ok	ST:00000D16	70 1A		strb	r2,[r3]	
ok	489	*return 2*x1;*				
ok	ST:00000D18	1D FB		add	r3,r7,#0x7	
ok	ST:00000D1A	78 1B		ldrb	r3,[r3]	
ok	ST:00000D1C	00 5B		lsl	r3,r3,#0x1	
partial	490	*}*				
ok	ST:00000D1E	1C 18		mov	r0,r3	
ok	ST:00000D20	46 BD		mov	r13,r7	
ok	ST:00000D22	B0 02		add	sp,#0x8	
ok	ST:00000D24	BC 80		pop	{r7}	

Coverage	Address	Code	Label	Mnemonics		Comment/Justification
ok	ST:00000D26	BC 02		pop	{r1}	
ok	ST:00000D28	47 08		bx	r1	
never	ST:00000D2A	46 C0		nop		
	491					
	492	*int func15(short x1) /* Parameter: 1 Word */*				
ok	493	*{*				
ok	ST:00000D2C	B5 80	func15:	push	{r7,r14}	
ok	ST:00000D2E	B0 82		sub	sp,#0x8	
ok	ST:00000D30	AF 00		add	r7,sp,#0x0	
ok	ST:00000D32	1C 02		mov	r2,r0	
ok	ST:00000D34	1D BB		add	r3,r7,#0x6	
ok	ST:00000D36	80 1A		strh	r2,[r3]	
ok	494	* return 2*x1;*				
ok	ST:00000D38	1D BB		add	r3,r7,#0x6	
ok	ST:00000D3A	22 00		mov	r2,#0x0	
ok	ST:00000D3C	5E 9B		ldrsh	r3,[r3,r2]	
ok	ST:00000D3E	00 5B		lsl	r3,r3,#0x1	
ok	495	*}*				
ok	ST:00000D40	1C 18		mov	r0,r3	
ok	ST:00000D42	46 BD		mov	r13,r7	
ok	ST:00000D44	B0 02		add	sp,#0x8	
ok	ST:00000D46	BC 80		pop	{r7}	
ok	ST:00000D48	BC 02		pop	{r1}	
ok	ST:00000D4A	47 08		bx	r1	
	496					
	497	*int func16(long x1) /* Parameter: 1 Long */*				
ok	498	*{*				
ok	ST:00000D4C	B5 80	func16:	push	{r7,r14}	
ok	ST:00000D4E	B0 82		sub	sp,#0x8	
ok	ST:00000D50	AF 00		add	r7,sp,#0x0	
ok	ST:00000D52	60 78		str	r0,[r7,#0x4]	
ok	499	* return (int) (2*x1);*				
ok	ST:00000D54	68 7B		ldr	r3,[r7,#0x4]	
ok	ST:00000D56	00 5B		lsl	r3,r3,#0x1	
ok	500	*}*				
ok	ST:00000D58	1C 18		mov	r0,r3	
ok	ST:00000D5A	46 BD		mov	r13,r7	
ok	ST:00000D5C	B0 02		add	sp,#0x8	
ok	ST:00000D5E	BC 80		pop	{r7}	
ok	ST:00000D60	BC 02		pop	{r1}	
ok	ST:00000D62	47 08		bx	r1	
	501					
	502	*int func17(short x1, short x2) /* Parameter: 2 Short */*				
ok	503	*{*				
ok	ST:00000D64	B5 80	func17:	push	{r7,r14}	
ok	ST:00000D66	B0 82		sub	sp,#0x8	
ok	ST:00000D68	AF 00		add	r7,sp,#0x0	
ok	ST:00000D6A	1C 0A		mov	r2,r1	
ok	ST:00000D6C	1D BB		add	r3,r7,#0x6	
ok	ST:00000D6E	1C 01		mov	r1,r0	
ok	ST:00000D70	80 19		strh	r1,[r3]	
ok	ST:00000D72	1D 3B		add	r3,r7,#0x4	
ok	ST:00000D74	80 1A		strh	r2,[r3]	
ok	504	* return x1*x2;*				
ok	ST:00000D76	1D BB		add	r3,r7,#0x6	
ok	ST:00000D78	21 00		mov	r1,#0x0	
ok	ST:00000D7A	5E 5B		ldrsh	r3,[r3,r1]	
ok	ST:00000D7C	1D 3A		add	r2,r7,#0x4	
ok	ST:00000D7E	21 00		mov	r1,#0x0	
ok	ST:00000D80	5E 52		ldrsh	r2,[r2,r1]	
ok	ST:00000D82	43 53		mul	r3,r2	
ok	505	*}*				
ok	ST:00000D84	1C 18		mov	r0,r3	
ok	ST:00000D86	46 BD		mov	r13,r7	
ok	ST:00000D88	B0 02		add	sp,#0x8	
ok	ST:00000D8A	BC 80		pop	{r7}	
ok	ST:00000D8C	BC 02		pop	{r1}	
ok	ST:00000D8E	47 08		bx	r1	

Coverage	Addr/Line	Code	Label	Mnemonics		Comment/Justification
	506					
	507	*int func18(short x1, long x2)*		*/* Parameter: Short,Long */*		
ok	508	*{*				
ok	ST:00000D90	B5 80	func18:	push	{r7,r14}	
ok	ST:00000D92	B0 82		sub	sp,#0x8	
ok	ST:00000D94	AF 00		add	r7,sp,#0x0	
ok	ST:00000D96	1C 02		mov	r2,r0	
ok	ST:00000D98	60 39		str	r1,[r7]	
ok	ST:00000D9A	1D BB		add	r3,r7,#0x6	
ok	ST:00000D9C	80 1A		strh	r2,[r3]	
ok	509	*return x1*x2;*				
ok	ST:00000D9E	1D BB		add	r3,r7,#0x6	
ok	ST:00000DA0	22 00		mov	r2,#0x0	
ok	ST:00000DA2	5E 9B		ldrsh	r3,[r3,r2]	
ok	ST:00000DA4	68 3A		ldr	r2,[r7]	
ok	ST:00000DA6	43 53		mul	r3,r2	
ok	510	*}*				
ok	ST:00000DA8	1C 18		mov	r0,r3	
ok	ST:00000DAA	46 BD		mov	r13,r7	
ok	ST:00000DAC	B0 02		add	sp,#0x8	
ok	ST:00000DAE	BC 80		pop	{r7}	
ok	ST:00000DB0	BC 02		pop	{r1}	
ok	ST:00000DB2	47 08		bx	r1	
	511					
	512	*int func19(long x1, short x2)*		*/* Parameter: Long,Short */*		
ok	513	*{*				
ok	ST:00000DB4	B5 80	func19:	push	{r7,r14}	
ok	ST:00000DB6	B0 82		sub	sp,#0x8	
ok	ST:00000DB8	AF 00		add	r7,sp,#0x0	
ok	ST:00000DBA	60 78		str	r0,[r7,#0x4]	
ok	ST:00000DBC	1C 0A		mov	r2,r1	
ok	ST:00000DBE	1C BB		add	r3,r7,#0x2	
ok	ST:00000DC0	80 1A		strh	r2,[r3]	
ok	514	*return x1*x2;*				
ok	ST:00000DC2	1C BB		add	r3,r7,#0x2	
ok	ST:00000DC4	22 00		mov	r2,#0x0	
ok	ST:00000DC6	5E 9B		ldrsh	r3,[r3,r2]	
ok	ST:00000DC8	68 7A		ldr	r2,[r7,#0x4]	
ok	ST:00000DCA	43 53		mul	r3,r2	
ok	515	*}*				
ok	ST:00000DCC	1C 18		mov	r0,r3	
ok	ST:00000DCE	46 BD		mov	r13,r7	
ok	ST:00000DD0	B0 02		add	sp,#0x8	
ok	ST:00000DD2	BC 80		pop	{r7}	
ok	ST:00000DD4	BC 02		pop	{r1}	
ok	ST:00000DD6	47 08		bx	r1	
	516					
	517	*int func20(short x1, short x2, short x3)*		*/* Parameter: 3 Short */*		
ok	518	*{*				
ok	ST:00000DD8	B5 80	func20:	push	{r7,r14}	
ok	ST:00000DDA	B0 82		sub	sp,#0x8	
ok	ST:00000DDC	AF 00		add	r7,sp,#0x0	
ok	ST:00000DDE	1D BB		add	r3,r7,#0x6	
ok	ST:00000DE0	80 18		strh	r0,[r3]	
ok	ST:00000DE2	1D 3B		add	r3,r7,#0x4	
ok	ST:00000DE4	80 19		strh	r1,[r3]	
ok	ST:00000DE6	1C BB		add	r3,r7,#0x2	
ok	ST:00000DE8	80 1A		strh	r2,[r3]	
ok	519	*return x1*x2*x3;*				
ok	ST:00000DEA	1D BB		add	r3,r7,#0x6	
ok	ST:00000DEC	21 00		mov	r1,#0x0	
ok	ST:00000DEE	5E 5B		ldrsh	r3,[r3,r1]	
ok	ST:00000DF0	1D 3A		add	r2,r7,#0x4	
ok	ST:00000DF2	21 00		mov	r1,#0x0	
ok	ST:00000DF4	5E 52		ldrsh	r2,[r2,r1]	
ok	ST:00000DF6	43 53		mul	r3,r2	
ok	ST:00000DF8	1C BA		add	r2,r7,#0x2	
ok	ST:00000DFA	21 00		mov	r1,#0x0	
ok	ST:00000DFC	5E 52		ldrsh	r2,[r2,r1]	

Coverage	Address	Code	Label	Mnemonics	Comment/Justification
ok	ST:00000DFE	1C 18		mov r3,r2	
ok	520	}			
ok	ST:00000E00	1C 18		mov r0,r3	
ok	ST:00000E02	46 BD		mov r13,r7	
ok	ST:00000E04	B0 02		add sp,#0x8	
ok	ST:00000E06	BC 80		pop {r7}	
ok	ST:00000E08	BC 02		pop {r1}	
ok	ST:00000E0A	47 08		bx r1	
	521				
	522	int func21(long x1, short x2, short x3) /* Parameter: Long,Short,Short */			
ok	523	{			
ok	ST:00000E0C	B5 80	func21:	push {r7,r14}	
ok	ST:00000E0E	B0 82		sub sp,#0x8	
ok	ST:00000E10	AF 00		add r7,sp,#0x0	
ok	ST:00000E12	60 78		str r0,[r7,#0x4]	
ok	ST:00000E14	1C BB		add r3,r7,#0x2	
ok	ST:00000E16	80 19		strh r1,[r3]	
ok	ST:00000E18	1C 3B		mov r3,r7	
ok	ST:00000E1A	80 1A		strh r2,[r3]	
ok	524	return x1*x2*x3;			
ok	ST:00000E1C	1C BB		add r3,r7,#0x2	
ok	ST:00000E1E	21 00		mov r1,#0x0	
ok	ST:00000E20	5E 5B		ldrsh r3,[r3,r1]	
ok	ST:00000E22	68 7A		ldr r2,[r7,#0x4]	
ok	ST:00000E24	43 53		mul r3,r2	
ok	ST:00000E26	1C 3A		mov r2,r7	
ok	ST:00000E28	21 00		mov r1,#0x0	
ok	ST:00000E2A	5E 52		ldrsh r2,[r2,r1]	
ok	ST:00000E2C	43 53		mul r3,r2	
partial	525	}			
ok	ST:00000E2E	1C 18		mov r0,r3	
ok	ST:00000E30	46 BD		mov r13,r7	
ok	ST:00000E32	B0 02		add sp,#0x8	
ok	ST:00000E34	BC 80		pop {r7}	
ok	ST:00000E36	BC 02		pop {r1}	
ok	ST:00000E38	47 08		bx r1	
never	ST:00000E3A	46 C0		nop	
	526				
	527	int func22(short x1, long x2, short x3) /* Parameter: Short,Long,Short */			
ok	528	{			
ok	ST:00000E3C	B5 80	func22:	push {r7,r14}	
ok	ST:00000E3E	B0 82		sub sp,#0x8	
ok	ST:00000E40	AF 00		add r7,sp,#0x0	
ok	ST:00000E42	60 39		str r1,[r7]	
ok	ST:00000E44	1D BB		add r3,r7,#0x6	
ok	ST:00000E46	1C 01		mov r1,r0	
ok	ST:00000E48	80 19		strh r1,[r3]	
ok	ST:00000E4A	1D 3B		add r3,r7,#0x4	
ok	ST:00000E4C	80 1A		strh r2,[r3]	
ok	529	return x1*x2*x3;			
ok	ST:00000E4E	1D BB		add r3,r7,#0x6	
ok	ST:00000E50	21 00		mov r1,#0x0	
ok	ST:00000E52	5E 5B		ldrsh r3,[r3,r1]	
ok	ST:00000E54	68 3A		ldr r2,[r7]	
ok	ST:00000E56	43 53		mul r3,r2	
ok	ST:00000E58	1D 3A		add r2,r7,#0x4	
ok	ST:00000E5A	21 00		mov r1,#0x0	
ok	ST:00000E5C	5E 52		ldrsh r2,[r2,r1]	
ok	ST:00000E5E	43 53		mul r3,r2	
ok	530	}			
ok	ST:00000E60	1C 18		mov r0,r3	
ok	ST:00000E62	46 BD		mov r13,r7	
ok	ST:00000E64	B0 02		add sp,#0x8	
ok	ST:00000E66	BC 80		pop {r7}	
ok	ST:00000E68	BC 02		pop {r1}	
ok	ST:00000E6A	47 08		bx r1	
	531				

Coverage	Addr/Line	Code	Label	Mnemonics	Comment/Justification
never	532	*int func23(short x1, short x2, long x3)*		/* Parameter: Short,Short,Long */	
never	533	*{*			
never	ST:00000E6C	B5 80	func23:	push	{r7,r14}
never	ST:00000E6E	B0 82		sub	sp,#0x8
never	ST:00000E70	AF 00		add	r7,sp,#0x0
never	ST:00000E72	60 3A		str	r2,[r7]
never	ST:00000E74	1D BB		add	r3,r7,#0x6
never	ST:00000E76	1C 02		mov	r2,r0
never	ST:00000E78	80 1A		strh	r2,[r3]
never	ST:00000E7A	1D 3B		add	r3,r7,#0x4
never	ST:00000E7C	1C 0A		mov	r2,r1
never	ST:00000E7E	80 1A		strh	r2,[r3]
never	534	*return x1*x2*x3;*			
never	ST:00000E80	1D BB		add	r3,r7,#0x6
never	ST:00000E82	21 00		mov	r1,#0x0
never	ST:00000E84	5E 5B		ldrsh	r3,[r3,r1]
never	ST:00000E86	1D 3A		add	r2,r7,#0x4
never	ST:00000E88	21 00		mov	r1,#0x0
never	ST:00000E8A	5E 52		ldrsh	r2,[r2,r1]
never	ST:00000E8C	43 53		mul	r3,r2
never	ST:00000E8E	68 3A		ldr	r2,[r7]
never	ST:00000E90	43 53		mul	r3,r2
never	535	*}*			
never	ST:00000E92	1C 18		mov	r0,r3
never	ST:00000E94	46 BD		mov	r13,r7
never	ST:00000E96	B0 02		add	sp,#0x8
never	ST:00000E98	BC 80		pop	{r7}
never	ST:00000E9A	BC 02		pop	{r1}
never	ST:00000E9C	47 08		bx	r1
never	ST:00000E9E	46 C0		nop	
	536				
	537	*char func24(void)*		/* Char Return */	
never	538	*{*			
never	ST:00000EA0	B5 80	func24:	push	{r7,r14}
never	ST:00000EA2	AF 00		add	r7,sp,#0x0
never	539	*return 55;*			
never	ST:00000EA4	23 37		mov	r3,#0x37
never	540	*}*			
never	ST:00000EA6	1C 18		mov	r0,r3
never	ST:00000EA8	46 BD		mov	r13,r7
never	ST:00000EAA	BC 80		pop	{r7}
never	ST:00000EAC	BC 02		pop	{r1}
never	ST:00000EAE	47 08		bx	r1
	541				
	542	*long func25(void)*		/* Long Return */	
never	543	*{*			
never	ST:00000EB0	B5 80	func25:	push	{r7,r14}
never	ST:00000EB2	AF 00		add	r7,sp,#0x0
never	544	*return 123456781;*			
never	ST:00000EB4	4B 02		ldr	r3,0xEC0
never	545	*}*			
never	ST:00000EB6	1C 18		mov	r0,r3
never	ST:00000EB8	46 BD		mov	r13,r7
never	ST:00000EBA	BC 80		pop	{r7}
never	ST:00000EBC	BC 02		pop	{r1}
never	ST:00000EBE	47 08		bx	r1
never	SP:00000EC0	00 BC 61 4E		dcd	0xBC614E
	546				
	547	*char * func26(void)*		/* Pointer Return */	
never	548	*{*			
never	ST:00000EC4	B5 80	func26:	push	{r7,r14}
never	ST:00000EC6	AF 00		add	r7,sp,#0x0
	549	*static char x1[] = "Babel Fish";*			
	550				
never	551	*return x1;*			
never	ST:00000EC8	4B 02		ldr	r3,0xED4
never	552	*}*			
never	ST:00000ECA	1C 18		mov	r0,r3

Coverage	Addr/Line	Code	Label	Mnemonics		Comment/Justification
never	ST:00000ECC	46 BD		mov	r13,r7	
never	ST:00000ECE	BC 80		pop	{r7}	
never	ST:00000ED0	BC 02		pop	{r1}	
never	ST:00000ED2	47 08		bx	r1	
never	SP:00000ED4	00 00 46 50		dcd	0x4650	
	553					
	554	*struct struct1 func27(short x1, short x2) /* function returning struct */*				
never	555	*{*				
never	ST:00000ED8	B5 90	func27:	push	{r4,r7,r14}	never called => dead code
never	ST:00000EDA	B0 83		sub	sp,#0x0C	
never	ST:00000EDC	AF 00		add	r7,sp,#0x0	
never	ST:00000EDE	60 78		str	r0,[r7,#0x4]	
never	ST:00000EE0	1C BB		add	r3,r7,#0x2	
never	ST:00000EE2	80 19		strh	r1,[r3]	
never	ST:00000EE4	1C 3B		mov	r3,r7	
never	ST:00000EE6	80 1A		strh	r2,[r3]	
never	556	* ast.count = x1*x2;*				
never	ST:00000EE8	1C BB		add	r3,r7,#0x2	
never	ST:00000EEA	20 00		mov	r0,#0x0	
never	ST:00000EEC	5E 1B		ldrsh	r3,[r3,r0]	
never	ST:00000EEE	1C 3A		mov	r2,r7	
never	ST:00000EF0	21 00		mov	r1,#0x0	
never	ST:00000EF2	5E 52		ldrsh	r2,[r2,r1]	
never	ST:00000EF4	43 5A		mul	r2,r3	
never	ST:00000EF6	4B 07		ldr	r3,0xF14	
never	ST:00000EF8	60 5A		str	r2,[r3,#0x4]	
	557					
never	558	* return ast;*				
never	ST:00000EFA	68 7B		ldr	r3,[r7,#0x4]	
never	ST:00000EFC	4A 05		ldr	r2,0xF14	
never	ST:00000EFE	CA 13		ldmia	r2!,{r0-r1,r4}	
never	ST:00000F00	C3 13		stmia	r3!,{r0-r1,r4}	
never	ST:00000F02	CA 11		ldmia	r2!,{r0,r4}	
never	ST:00000F04	C3 11		stmia	r3!,{r0,r4}	
never	559	*}*				
never	ST:00000F06	68 78		ldr	r0,[r7,#0x4]	
never	ST:00000F08	46 BD		mov	r13,r7	
never	ST:00000F0A	B0 03		add	sp,#0x0C	
never	ST:00000F0C	BC 90		pop	{r4,r7}	
never	ST:00000F0E	BC 02		pop	{r1}	
never	ST:00000F10	47 08		bx	r1	
never	ST:00000F12	46 C0		nop		
never	SP:00000F14	00 00 47 50		dcd	0x4750	
	560					
	561					
	562	*#if !defined(__aarch32__)&&!defined(__thumb__)&&!defined(__thumb2__)&&!defined(__aarch64__)*				
	563	*# define __aarch32__ 1*				
	564	*#endif*				
	565					
	566	*static int test_cond_instr(int arg1, int arg2) /* function with conditional (non-branch) instructions */*				
never	567	*{*				
never	ST:00000F18	B5 80	test_cond_instr:	push	{r7,r14}	
never	ST:00000F1A	B0 84		sub	sp,#0x10	
never	ST:00000F1C	AF 00		add	r7,sp,#0x0	
never	ST:00000F1E	60 78		str	r0,[r7,#0x4]	
never	ST:00000F20	60 39		str	r1,[r7]	
		...				
	572	* "ittte ne \n\t"*				
	573	* "mulne %[arg2],%[arg1],%[arg2] \n\t"*				
	574	* "addne %[arg2],%[arg2],#42 \n\t"*				
	575	* "asrne %[result],%[arg2],#1 \n\t"*				
	576	* "moveq %[result],#0 \n\t"*				
	577	* : [result] "=r" (result) : [arg1] "r" (arg1), [arg2] "r" (arg2)*				
	578	*);*				
	579	*#else*				
never	580	* arg2 = arg1 - arg2;*				
never	ST:00000F22	68 7A		ldr	r2,[r7,#0x4]	
never	ST:00000F24	68 3B		ldr	r3,[r7]	

Coverage	Addr/Line	Code	Label	Mnemonics	Comment/Justification
never	ST:00000F26	7A 18		sub	r3,r2,r3
never	ST:00000F28	60 3B		str	r3,[r7]
never	581	*if (arg2){*			
never	ST:00000F2A	68 3B		ldr	r3,[r7]
never	ST:00000F2C	2B 00		cmp	r3,#0x0
never	ST:00000F2E	D0 0A		beq	0xF46
never	582	*arg2 = arg2 * arg1;*			
never	ST:00000F30	68 3B		ldr	r3,[r7]
never	ST:00000F32	68 7A		ldr	r2,[r7,#0x4]
never	ST:00000F34	43 53		mul	r3,r2
never	ST:00000F36	60 3B		str	r3,[r7]
never	583	*arg2 = arg2 + 42;*			
never	ST:00000F38	68 3B		ldr	r3,[r7]
never	ST:00000F3A	33 2A		add	r3,#0x2A
never	ST:00000F3C	60 3B		str	r3,[r7]
never	584	*result = arg2 >> 1;*			
never	ST:00000F3E	68 3B		ldr	r3,[r7]
never	ST:00000F40	10 5B		asr	r3,r3,#0x1
never	ST:00000F42	60 FB		str	r3,[r7,#0x0C]
never	ST:00000F44	E0 01		b	0xF4A
	585	*} else {*			
never	586	*result = 0;*			
never	ST:00000F46	23 00		mov	r3,#0x0
never	ST:00000F48	60 FB		str	r3,[r7,#0x0C]
	587	*}*			
	588	*#endif*			
never	589	*return result;*			
never	ST:00000F4A	68 FB		ldr	r3,[r7,#0x0C]
never	590	*}*			
never	ST:00000F4C	1C 18		mov	r0,r3
never	ST:00000F4E	46 BD		mov	r13,r7
never	ST:00000F50	B0 04		add	sp,#0x10
never	ST:00000F52	BC 80		pop	{r7}
never	ST:00000F54	BC 02		pop	{r1}
never	ST:00000F56	47 08		bx	r1
	591				
	592				
	593	*double sinewave[628];*			
	594				
	595	*void func_sin(void)*			
ok	596	*{*			
ok	ST:00000F58	B5 F0	**func_sin:**	push	{r4-r7,r14}
ok	ST:00000F5A	B0 85		sub	sp,#0x14
ok	ST:00000F5C	AF 00		add	r7,sp,#0x0
	597	*int index;*			
	598	*double x;*			
	599				
ok	600	*index = 0;*			
ok	ST:00000F5E	23 00		mov	r3,#0x0
ok	ST:00000F60	60 FB		str	r3,[r7,#0x0C]
ok	601	*for (x = 0.0 ; x < 62.8 ; x += 0.1)*			
ok	ST:00000F62	4C 2C		ldr	r4,0x1014
ok	ST:00000F64	4B 2A		ldr	r3,0x1010
ok	ST:00000F66	60 3B		str	r3,[r7]
ok	ST:00000F68	60 7C		str	r4,[r7,#0x4]
ok	ST:00000F6A	E0 3A		b	0xFE2
ok	602	*sinewave[index++] = (sin(x)/(x+0.1)) * 300.0 + 80.0;*			
ok	ST:00000F6C	68 3B		ldr	r3,[r7]
ok	ST:00000F6E	68 7C		ldr	r4,[r7,#0x4]
ok	ST:00000F70	1C 18		mov	r0,r3
ok	ST:00000F72	1C 21		mov	r1,r4
ok	ST:00000F74	F0 01 F8 C0		bl	0x20F8
ok	ST:00000F78	1C 05		mov	r5,r0
ok	ST:00000F7A	1C 0E		mov	r6,r1
ok	ST:00000F7C	68 38		ldr	r0,[r7]
ok	ST:00000F7E	68 79		ldr	r1,[r7,#0x4]
ok	ST:00000F80	4A 25		ldr	r2,0x1018
ok	ST:00000F82	4B 26		ldr	r3,0x101C

Coverage	Address	Code	Label	Mnemonic	Comment/Justification
ok	ST:00000F84	F0 03 F9 DC		bl 0x4340	
ok	ST:00000F88	1C 03		mov r3,r0	
ok	ST:00000F8A	1C 0C		mov r4,r1	
ok	ST:00000F8C	1C 28		mov r0,r5	
ok	ST:00000F8E	1C 31		mov r1,r6	
ok	ST:00000F90	1C 1A		mov r2,r3	
ok	ST:00000F92	1C 23		mov r3,r4	
ok	ST:00000F94	F0 03 F9 AC		bl 0x42F0	
ok	ST:00000F98	1C 03		mov r3,r0	
ok	ST:00000F9A	1C 0C		mov r4,r1	
ok	ST:00000F9C	1C 18		mov r0,r3	
ok	ST:00000F9E	1C 21		mov r1,r4	
ok	ST:00000FA0	4A 1F		ldr r2,0x1020	
ok	ST:00000FA2	4B 20		ldr r3,0x1024	
ok	ST:00000FA4	F0 03 F9 B8		bl 0x4318	
ok	ST:00000FA8	1C 03		mov r3,r0	
ok	ST:00000FAA	1C 0C		mov r4,r1	
ok	ST:00000FAC	1C 18		mov r0,r3	
ok	ST:00000FAE	1C 21		mov r1,r4	
ok	ST:00000FB0	4A 1D		ldr r2,0x1028	
ok	ST:00000FB2	4B 1E		ldr r3,0x102C	
ok	ST:00000FB4	F0 03 F9 C4		bl 0x4340	
ok	ST:00000FB8	1C 03		mov r3,r0	
ok	ST:00000FBA	1C 0C		mov r4,r1	
ok	ST:00000FBC	49 1E		ldr r1,0x1038	
ok	ST:00000FBE	68 FA		ldr r2,[r7,#0x0C]	
ok	ST:00000FC0	00 D2		lsl r2,r2,#0x3	
ok	ST:00000FC2	18 8A		add r2,r1,r2	
ok	ST:00000FC4	60 13		str r3,[r2]	
ok	ST:00000FC6	60 54		str r4,[r2,#0x4]	
ok	ST:00000FC8	68 FB		ldr r3,[r7,#0x0C]	
ok	ST:00000FCA	33 01		add r3,#0x1	
ok	ST:00000FCC	60 FB		str r3,[r7,#0x0C]	
ok	601	*for (x = 0.0 ; x < 62.8 ; x += 0.1)*			
ok	ST:00000FCE	68 38		ldr r0,[r7]	
ok	ST:00000FD0	68 79		ldr r1,[r7,#0x4]	
ok	ST:00000FD2	4A 11		ldr r2,0x1018	
ok	ST:00000FD4	4B 11		ldr r3,0x101C	
ok	ST:00000FD6	F0 03 F9 B3		bl 0x4340	
ok	ST:00000FDA	1C 03		mov r3,r0	
ok	ST:00000FDC	1C 0C		mov r4,r1	
ok	ST:00000FDE	60 3B		str r3,[r7]	
ok	ST:00000FE0	60 7C		str r4,[r7,#0x4]	
ok	ST:00000FE2	23 01		mov r3,#0x1	
ok	ST:00000FE4	1C 1C		mov r4,r3	
ok	ST:00000FE6	68 38		ldr r0,[r7]	
ok	ST:00000FE8	68 79		ldr r1,[r7,#0x4]	
ok	ST:00000FEA	4A 11		ldr r2,0x1030	
ok	ST:00000FEC	4B 11		ldr r3,0x1034	
ok	ST:00000FEE	F0 03 F9 8F		bl 0x4310	
ok	ST:00000FF2	1C 03		mov r3,r0	
ok	ST:00000FF4	D1 01		bne 0xFFA	
ok	ST:00000FF6	23 00		mov r3,#0x0	
ok	ST:00000FF8	1C 1C		mov r4,r3	
ok	ST:00000FFA	06 23		lsl r3,r4,#0x18	
ok	ST:00000FFC	0E 1B		lsr r3,r3,#0x18	
ok	ST:00000FFE	D1 B5		bne 0xF6C	
partial	603	*}*			
ok	ST:00001000	46 BD		mov r13,r7	
ok	ST:00001002	B0 05		add sp,#0x14	
ok	ST:00001004	BC F0		pop {r4-r7}	
ok	ST:00001006	BC 01		pop {r0}	
ok	ST:00001008	47 00		bx r0	
never	ST:0000100A	46 C0		nop	
never	ST:0000100C	46 C0		nop	
never	ST:0000100E	46 C0		nop	
read	SP:00001010	00 00 00 00		dcd 0x0	
read	SP:00001014	00 00 00 00		dcd 0x0	

Coverage	Addr/Line	Code	Label	Mnemonics		Comment/Justification
read	SP:00001018	99 99 99 9A		dcd	0x9999999A	
read	SP:0000101C	3F B9 99 99		dcd	0x3FB99999	
read	SP:00001020	00 00 00 00		dcd	0x0	
read	SP:00001024	40 72 C0 00		dcd	0x4072C000	
read	SP:00001028	00 00 00 00		dcd	0x0	
read	SP:0000102C	40 54 00 00		dcd	0x40540000	
read	SP:00001030	66 66 66 66		dcd	0x66666666	
read	SP:00001034	40 4F 66 66		dcd	0x404F6666	
read	SP:00001038	00 00 49 F0		dcd	0x49F0	
	604					
	605	*void init_linked_list(void)*				
ok	606	*{*				
ok	ST:0000103C	B5 80	init_linked_list:	push	{r7,r14}	
ok	ST:0000103E	B0 82		sub	sp,#0x8	
ok	ST:00001040	AF 00		add	r7,sp,#0x0	
ok	607	*int i = 0;*				
ok	ST:00001042	23 00		mov	r3,#0x0	
ok	ST:00001044	60 7B		str	r3,[r7,#0x4]	
ok	608	*const int last = sizeof(pLinkedListBuf) / sizeof(pLinkedListBuf[0]) - 1;*				
ok	ST:00001046	23 04		mov	r3,#0x4	
ok	ST:00001048	60 3B		str	r3,[r7]	
	609	*static const char* const text[] = {"Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"};*				
	610					
ok	611	*ast.left = pLinkedListBuf;*				
ok	ST:0000104A	4B 32		ldr	r3,0x1114	
ok	ST:0000104C	4A 32		ldr	r2,0x1118	
ok	ST:0000104E	60 9A		str	r2,[r3,#0x8]	
ok	612	*ast.right = NULL;*				
ok	ST:00001050	4B 30		ldr	r3,0x1114	
ok	ST:00001052	22 00		mov	r2,#0x0	
ok	ST:00001054	60 DA		str	r2,[r3,#0x0C]	
ok	613	*ast.count = 0;*				
ok	ST:00001056	4B 2F		ldr	r3,0x1114	
ok	ST:00001058	22 00		mov	r2,#0x0	
ok	ST:0000105A	60 5A		str	r2,[r3,#0x4]	
ok	614	*ast.word = NULL;*				
ok	ST:0000105C	4B 2D		ldr	r3,0x1114	
ok	ST:0000105E	22 00		mov	r2,#0x0	
ok	ST:00001060	60 1A		str	r2,[r3]	
	615					
ok	616	*for (i = 0; i <= last; i++) {*				
ok	ST:00001062	23 00		mov	r3,#0x0	
ok	ST:00001064	60 7B		str	r3,[r7,#0x4]	
ok	ST:00001066	E0 4C		b	0x1102	
ok	617	*pLinkedListBuf[i].left = (i==last) ? NULL : &pLinkedListBuf[i+1]; /* next element */*				
ok	ST:00001068	68 7A		ldr	r2,[r7,#0x4]	
ok	ST:0000106A	68 3B		ldr	r3,[r7]	
ok	ST:0000106C	42 9A		cmp	r2,r3	
ok	ST:0000106E	D0 09		beq	0x1084	
ok	ST:00001070	68 7B		ldr	r3,[r7,#0x4]	
ok	ST:00001072	1C 5A		add	r2,r3,#0x1	
ok	ST:00001074	1C 13		mov	r3,r2	
ok	ST:00001076	00 9B		lsl	r3,r3,#0x2	
ok	ST:00001078	18 9B		add	r3,r3,r2	
ok	ST:0000107A	00 9B		lsl	r3,r3,#0x2	
ok	ST:0000107C	1C 1A		mov	r2,r3	
ok	ST:0000107E	4B 26		ldr	r3,0x1118	
ok	ST:00001080	18 D2		add	r2,r2,r3	
ok	ST:00001082	E0 00		b	0x1086	
ok	ST:00001084	22 00		mov	r2,#0x0	
ok	ST:00001086	48 24		ldr	r0,0x1118	
ok	ST:00001088	68 79		ldr	r1,[r7,#0x4]	
ok	ST:0000108A	1C 0B		mov	r3,r1	
ok	ST:0000108C	00 9B		lsl	r3,r3,#0x2	
ok	ST:0000108E	18 5B		add	r3,r3,r1	
ok	ST:00001090	00 9B		lsl	r3,r3,#0x2	
ok	ST:00001092	18 C3		add	r3,r0,r3	

Coverage	Addr/Line	Code	Label	Mnemonics	Comment/Justification
ok	ST:00001094	23 08		add r3,#0x8	
ok	ST:00001096	60 1A		str r2,[r3]	
ok	618		pLinkedListBuf[i].right = (i==0) ? &ast : &pLinkedListBuf[i-1]; /* previous element */		
ok	ST:00001098	68 7B		ldr r3,[r7,#0x4]	
ok	ST:0000109A	2B 00		cmp r3,#0x0	
ok	ST:0000109C	D0 09		beq 0x10B2	
ok	ST:0000109E	68 7B		ldr r3,[r7,#0x4]	
ok	ST:000010A0	1E 5A		sub r2,r3,#0x1	
ok	ST:000010A2	1C 13		mov r3,r2	
ok	ST:000010A4	00 9B		lsl r3,r3,#0x2	
ok	ST:000010A6	18 9B		add r3,r3,r2	
ok	ST:000010A8	00 9B		lsl r3,r3,#0x2	
ok	ST:000010AA	1C 1A		mov r2,r3	
ok	ST:000010AC	4B 1A		ldr r3,0x1118	
ok	ST:000010AE	18 D2		add r2,r2,r3	
ok	ST:000010B0	E0 00		b 0x10B4	
ok	ST:000010B2	4A 18		ldr r2,0x1114	
ok	ST:000010B4	48 18		ldr r0,0x1118	
ok	ST:000010B6	68 79		ldr r1,[r7,#0x4]	
ok	ST:000010B8	1C 0B		mov r3,r1	
ok	ST:000010BA	00 9B		lsl r3,r3,#0x2	
ok	ST:000010BC	18 5B		add r3,r3,r1	
ok	ST:000010BE	00 9B		lsl r3,r3,#0x2	
ok	ST:000010C0	18 C3		add r3,r0,r3	
ok	ST:000010C2	33 0C		add r3,#0x0C	
ok	ST:000010C4	60 1A		str r2,[r3]	
ok	619		pLinkedListBuf[i].count = i;		
ok	ST:000010C6	49 14		ldr r1,0x1118	
ok	ST:000010C8	68 7A		ldr r2,[r7,#0x4]	
ok	ST:000010CA	1C 13		mov r3,r2	
ok	ST:000010CC	00 9B		lsl r3,r3,#0x2	
ok	ST:000010CE	18 9B		add r3,r3,r2	
ok	ST:000010D0	00 9B		lsl r3,r3,#0x2	
ok	ST:000010D2	18 CB		add r3,r1,r3	
ok	ST:000010D4	33 04		add r3,#0x4	
ok	ST:000010D6	68 7A		ldr r2,[r7,#0x4]	
ok	ST:000010D8	60 1A		str r2,[r3]	
ok	620		pLinkedListBuf[i].word = text[i % 7];		
ok	ST:000010DA	68 7B		ldr r3,[r7,#0x4]	
ok	ST:000010DC	1C 18		mov r0,r3	
ok	ST:000010DE	21 07		mov r1,#0x7	
ok	ST:000010E0	F0 03 F9 26		bl 0x4330	
ok	ST:000010E4	1C 0B		mov r3,r1	
ok	ST:000010E6	1C 1A		mov r2,r3	
ok	ST:000010E8	4B 0C		ldr r3,0x111C	
ok	ST:000010EA	00 92		lsl r2,r2,#0x2	
ok	ST:000010EC	58 D0		ldr r0,[r2,r3]	
ok	ST:000010EE	49 0A		ldr r1,0x1118	
ok	ST:000010F0	68 7A		ldr r2,[r7,#0x4]	
ok	ST:000010F2	1C 13		mov r3,r2	
ok	ST:000010F4	00 9B		lsl r3,r3,#0x2	
ok	ST:000010F6	18 9B		add r3,r3,r2	
ok	ST:000010F8	00 9B		lsl r3,r3,#0x2	
ok	ST:000010FA	50 58		str r0,[r3,r1]	
	615				
ok	616		for (i = 0; i <= last; i++) {		
ok	ST:000010FC	68 7B		ldr r3,[r7,#0x4]	
ok	ST:000010FE	33 01		add r3,#0x1	
ok	ST:00001100	60 7B		str r3,[r7,#0x4]	
ok	ST:00001102	68 7A		ldr r2,[r7,#0x4]	
ok	ST:00001104	68 3B		ldr r3,[r7]	
ok	ST:00001106	42 9A		cmp r2,r3	
ok	ST:00001108	DD AE		ble 0x1068	
	621		}		
partial	622		}		
ok	ST:0000110A	46 BD		mov r13,r7	
ok	ST:0000110C	B0 02		add sp,#0x8	

Coverage	Address	Code	Label	Mnemonic		Comment/Justification
ok	ST:0000110E	BC 80		pop	{r7}	
ok	ST:00001110	BC 01		pop	{r0}	
ok	ST:00001112	47 00		bx	r0	
read	SP:00001114	00 00 47 50		dcd	0x4750	
read	SP:00001118	00 00 47 64		dcd	0x4764	
read	SP:0000111C	00 00 44 28		dcd	0x4428	
	623					
	624	*static char subst(char c)*				
never	625	*{*				
never	ST:00001120	B5 80	subst:	push	{r7,r14}	
never	ST:00001122	B0 82		sub	sp,#0x8	
never	ST:00001124	AF 00		add	r7,sp,#0x0	
never	ST:00001126	1C 02		mov	r2,r0	
never	ST:00001128	1D FB		add	r3,r7,#0x7	
never	ST:0000112A	70 1A		strb	r2,[r3]	
never	626	*switch (c){*				
never	ST:0000112C	1D FB		add	r3,r7,#0x7	
never	ST:0000112E	78 1B		ldrb	r3,[r3]	
never	ST:00001130	3B 61		sub	r3,#0x61	
never	ST:00001132	2B 14		cmp	r3,#0x14	
never	ST:00001134	D8 0E		bhi	0x1154	
never	ST:00001136	00 9A		lsl	r2,r3,#0x2	
never	ST:00001138	4B 0A		ldr	r3,0x1164	
never	ST:0000113A	18 D3		add	r3,r2,r3	
never	ST:0000113C	68 1B		ldr	r3,[r3]	
never	ST:0000113E	46 9F		mov	pc,r3	
	627	*case 'a':*				
never	628	*return 'e';*				
never	ST:00001140	23 65		mov	r3,#0x65	
never	ST:00001142	E0 09		b	0x1158	
	629	*case 'e':*				
never	630	*return 'i';*				
never	ST:00001144	23 69		mov	r3,#0x69	
never	ST:00001146	E0 07		b	0x1158	
	631	*case 'i':*				
never	632	*return 'u';*				
never	ST:00001148	23 75		mov	r3,#0x75	
never	ST:0000114A	E0 05		b	0x1158	
	633	*case 'o':*				
never	634	*return 'a';*				
never	ST:0000114C	23 61		mov	r3,#0x61	
never	ST:0000114E	E0 03		b	0x1158	
	635	*case 'u':*				
never	636	*return 'o';*				
never	ST:00001150	23 6F		mov	r3,#0x6F	
never	ST:00001152	E0 01		b	0x1158	
	637	*}*				
never	638	*return c;*				
never	ST:00001154	1D FB		add	r3,r7,#0x7	
never	ST:00001156	78 1B		ldrb	r3,[r3]	
never	639	*}*				
never	ST:00001158	1C 18		mov	r0,r3	
never	ST:0000115A	46 BD		mov	r13,r7	
never	ST:0000115C	B0 02		add	sp,#0x8	
never	ST:0000115E	BC 80		pop	{r7}	
never	ST:00001160	BC 02		pop	{r1}	
never	ST:00001162	47 08		bx	r1	
never	SP:00001164	00 00 43 90		dcd	0x4390	
	640					
	641	*static char* encode(char str[])*				
never	642	*{*				
never	ST:00001168	B5 90	encode:	push	{r4,r7,r14}	
never	ST:0000116A	B0 85		sub	sp,#0x14	
never	ST:0000116C	AF 00		add	r7,sp,#0x0	
never	ST:0000116E	60 78		str	r0,[r7,#0x4]	
	643	*int i;*				
	644					
never	645	*for (i = 0; str[i]; i++){*				
never	ST:00001170	23 00		mov	r3,#0x0	

Coverage	Addr/Line	Code	Label	Mnemonics		Comment/Justification
never	ST:00001172	68 FB		ldr	r3,[r7,#0x0C]	
never	ST:00001174	E0 3D		b	0x11F2	
never	646		*char c = str[i];*			
never	ST:00001176	68 FB		ldr	r3,[r7,#0x0C]	
never	ST:00001178	68 7A		ldr	r2,[r7,#0x4]	
never	ST:0000117A	18 D2		add	r2,r2,r3	
never	ST:0000117C	1C 3B		mov	r3,r7	
never	ST:0000117E	33 0B		add	r3,#0x0B	
never	ST:00001180	78 12		ldrb	r2,[r2]	
never	ST:00001182	70 1A		strb	r2,[r3]	
never	647		*if ('A' <= c && c <= 'Z'){*			
never	ST:00001184	1C 3B		mov	r3,r7	
never	ST:00001186	33 0B		add	r3,#0x0B	
never	ST:00001188	78 1B		ldrb	r3,[r3]	
never	ST:0000118A	2B 40		cmp	r3,#0x40	
never	ST:0000118C	D9 1D		bls	0x11CA	
never	ST:0000118E	1C 3B		mov	r3,r7	
never	ST:00001190	33 0B		add	r3,#0x0B	
never	ST:00001192	78 1B		ldrb	r3,[r3]	
never	ST:00001194	2B 5A		cmp	r3,#0x5A	
never	ST:00001196	D8 18		bhi	0x11CA	
never	648		*c = c + ('a'-'A');*			
never	ST:00001198	1C 3B		mov	r3,r7	
never	ST:0000119A	33 0B		add	r3,#0x0B	
never	ST:0000119C	1C 3A		mov	r2,r7	
never	ST:0000119E	32 0B		add	r2,#0x0B	
never	ST:000011A0	78 12		ldrb	r2,[r2]	
never	ST:000011A2	32 20		add	r2,#0x20	
never	ST:000011A4	70 1A		strb	r2,[r3]	
never	649		*c = subst(c);*			
never	ST:000011A6	1C 3C		mov	r4,r7	
never	ST:000011A8	34 0B		add	r4,#0x0B	
never	ST:000011AA	1C 3B		mov	r3,r7	
never	ST:000011AC	33 0B		add	r3,#0x0B	
never	ST:000011AE	78 1B		ldrb	r3,[r3]	
never	ST:000011B0	1C 18		mov	r0,r3	
never	ST:000011B2	F7 FF FF B5		bl	0x1120	
never	ST:000011B6	1C 03		mov	r3,r0	
never	ST:000011B8	70 23		strb	r3,[r4]	
never	650		*c = c - ('a'-'A');*			
never	ST:000011BA	1C 3B		mov	r3,r7	
never	ST:000011BC	33 0B		add	r3,#0x0B	
never	ST:000011BE	1C 3A		mov	r2,r7	
never	ST:000011C0	32 0B		add	r2,#0x0B	
never	ST:000011C2	78 12		ldrb	r2,[r2]	
never	ST:000011C4	3A 20		sub	r2,#0x20	
never	ST:000011C6	70 1A		strb	r2,[r3]	
never	ST:000011C8	E0 09		b	0x11DE	
	651		*} else {*			
never	652		*c = subst(c);*			
never	ST:000011CA	1C 3C		mov	r4,r7	
never	ST:000011CC	34 0B		add	r4,#0x0B	
never	ST:000011CE	1C 3B		mov	r3,r7	
never	ST:000011D0	33 0B		add	r3,#0x0B	
never	ST:000011D2	78 1B		ldrb	r3,[r3]	
never	ST:000011D4	1C 18		mov	r0,r3	
never	ST:000011D6	F7 FF FF A3		bl	0x1120	
never	ST:000011DA	1C 03		mov	r3,r0	
never	ST:000011DC	70 23		strb	r3,[r4]	
	653		*}*			
never	654		*str[i] = c;*			
never	ST:000011DE	68 FB		ldr	r3,[r7,#0x0C]	
never	ST:000011E0	68 7A		ldr	r2,[r7,#0x4]	
never	ST:000011E2	18 D3		add	r3,r2,r3	
never	ST:000011E4	1C 3A		mov	r2,r7	
never	ST:000011E6	32 0B		add	r2,#0x0B	
never	ST:000011E8	78 12		ldrb	r2,[r2]	
never	ST:000011EA	70 1A		strb	r2,[r3]	

Coverage	Addr/Line	Code	Label	Mnemonics	Comment/Justification
	643	int i;			
	644				
never	645	for (i = 0; str[i]; i++){			
never	ST:000011EC	68 FB		ldr r3,[r7,#0x0C]	
never	ST:000011EE	33 01		add r3,#0x1	
never	ST:000011F0	60 FB		str r3,[r7,#0x0C]	
never	ST:000011F2	68 FB		ldr r3,[r7,#0x0C]	
never	ST:000011F4	68 7A		ldr r2,[r7,#0x4]	
never	ST:000011F6	18 D3		add r3,r2,r3	
never	ST:000011F8	78 1B		ldrb r3,[r3]	
never	ST:000011FA	2B 00		cmp r3,#0x0	
never	ST:000011FC	D1 BB		bne 0x1176	
	655	}			
never	656	return str;			
never	ST:000011FE	68 7B		ldr r3,[r7,#0x4]	
never	657	}			
never	ST:00001200	1C 18		mov r0,r3	
never	ST:00001202	46 BD		mov r13,r7	
never	ST:00001204	B0 05		add sp,#0x14	
never	ST:00001206	BC 90		pop {r4,r7}	
never	ST:00001208	BC 02		pop {r1}	
never	ST:0000120A	47 08		bx r1	
		...			
	661	{			
	662	short x;			
	663	short y;			
	664	};			
	665				
	666	struct shortrecord vshortrecord;			
	667				
	668	struct shortrecord func28(struct shortrecord x)			
never	669	{			
never	ST:0000120C	B5 80	func28:	push {r7,r14}	
never	ST:0000120E	B0 84		sub sp,#0x10	
never	ST:00001210	AF 00		add r7,sp,#0x0	
never	ST:00001212	1D 3B		add r3,r7,#0x4	
never	ST:00001214	60 18		str r0,[r3]	
never	670	x.x++;			
never	ST:00001216	1D 3B		add r3,r7,#0x4	
never	ST:00001218	88 1B		ldrh r3,[r3]	
never	ST:0000121A	33 01		add r3,#0x1	
never	ST:0000121C	04 1B		lsl r3,r3,#0x10	
never	ST:0000121E	0C 1A		lsr r2,r3,#0x10	
never	ST:00001220	1D 3B		add r3,r7,#0x4	
never	ST:00001222	80 1A		strh r2,[r3]	
never	671	x.y = (short) (x.x+1);			
never	ST:00001224	1D 3B		add r3,r7,#0x4	
never	ST:00001226	88 1B		ldrh r3,[r3]	
never	ST:00001228	04 1B		lsl r3,r3,#0x10	
never	ST:0000122A	0C 1B		lsr r3,r3,#0x10	
never	ST:0000122C	33 01		add r3,#0x1	
never	ST:0000122E	04 1B		lsl r3,r3,#0x10	
never	ST:00001230	0C 1B		lsr r3,r3,#0x10	
never	ST:00001232	04 1B		lsl r3,r3,#0x10	
never	ST:00001234	0C 1A		lsr r2,r3,#0x10	
never	ST:00001236	1D 3B		add r3,r7,#0x4	
never	ST:00001238	80 5A		strh r2,[r3,#0x2]	
never	672	return x;			
never	ST:0000123A	1C 3B		mov r3,r7	
never	ST:0000123C	33 0C		add r3,#0x0C	
never	ST:0000123E	1D 3A		add r2,r7,#0x4	
never	ST:00001240	68 12		ldr r2,[r2]	
never	ST:00001242	60 1A		str r2,[r3]	
never	ST:00001244	1C 3A		mov r2,r7	
never	ST:00001246	32 0C		add r2,#0x0C	
never	ST:00001248	23 00		mov r3,#0x0	
never	ST:0000124A	88 11		ldrh r1,[r2]	
never	ST:0000124C	04 09		lsl r1,r1,#0x10	

Coverage	Address	Code	Label	Mnemonic	Operands	Comment/Justification
never	ST:0000124E	0C 09		lsr	r1,r1,#0x10	
never	ST:00001250	0C 1B		lsr	r3,r3,#0x10	
never	ST:00001252	04 1B		lsl	r3,r3,#0x10	
never	ST:00001254	43 0B		orr	r3,r1	
never	ST:00001256	88 52		ldrh	r2,[r2,#0x2]	
never	ST:00001258	04 12		lsl	r2,r2,#0x10	
never	ST:0000125A	04 1B		lsl	r3,r3,#0x10	
never	ST:0000125C	0C 1B		lsr	r3,r3,#0x10	
never	ST:0000125E	43 13		orr	r3,r2	
never	673	}				
never	ST:00001260	1C 18		mov	r0,r3	
never	ST:00001262	46 BD		mov	r13,r7	
never	ST:00001264	B0 04		add	sp,#0x10	
never	ST:00001266	BC 80		pop	{r7}	
never	ST:00001268	BC 02		pop	{r1}	
never	ST:0000126A	47 08		bx	r1	
		...				
	678	unsigned long l[1];				
	679	} datas;				
	680					
	681	static int sieve(void);				
	682					
	683	void (*monHook)(void) __attribute__ ((section (".data"))) = 0;				
	684					
	685	int main(void)				
ok	686	{				
ok	ST:0000126C	B5 F0	main:	push	{r4-r7,r14}	
ok	ST:0000126E	B0 8D		sub	sp,#0x34	
ok	ST:00001270	AF 02		add	r7,sp,#0x8	
	687	int j;				
	688	short int inc, sign;				
	689	char *p;				
	690					
ok	691	func_sin();				
ok	ST:00001272	F7 FF FE 71		bl	0xF58	
	692					
	693	do {				
taken	694	if (monHook)				
ok	ST:00001276	4B BE		ldr	r3,0x1570	
ok	ST:00001278	68 1B		ldr	r3,[r3]	
ok	ST:0000127A	2B 00		cmp	r3,#0x0	
taken	ST:0000127C	D0 03		beq	0x1286	
never	695	monHook();				
never	ST:0000127E	4B BC		ldr	r3,0x1570	
never	ST:00001280	68 1B		ldr	r3,[r3]	
never	ST:00001282	F0 00 F9 FB		bl	0x167C	
	696					
ok	697	mstatic1 = 12;				
ok	ST:00001286	4B BB		ldr	r3,0x1574	
ok	ST:00001288	22 0C		mov	r2,#0x0C	
ok	ST:0000128A	60 1A		str	r2,[r3]	
ok	698	mstatic2 = 34;				
ok	ST:0000128C	4B BA		ldr	r3,0x1578	
ok	ST:0000128E	22 22		mov	r2,#0x22	
ok	ST:00001290	60 1A		str	r2,[r3]	
ok	699	mcount++;				
ok	ST:00001292	4B BA		ldr	r3,0x157C	
ok	ST:00001294	68 1B		ldr	r3,[r3]	
ok	ST:00001296	1C 5A		add	r2,r3,#0x1	
ok	ST:00001298	4B B8		ldr	r3,0x157C	
ok	ST:0000129A	60 1A		str	r2,[r3]	
	700					
ok	701	inc = (4 * 15000) / period;				
ok	ST:0000129C	4B B8		ldr	r3,0x1580	
ok	ST:0000129E	68 1B		ldr	r3,[r3]	
ok	ST:000012A0	48 B8		ldr	r0,0x1584	
ok	ST:000012A2	1C 19		mov	r1,r3	
ok	ST:000012A4	F0 03 F8 40		bl	0x4328	
ok	ST:000012A8	1C 03		mov	r3,r0	

Coverage	Address	Code	Label	Mnemonics		Comment/Justification
ok	ST:000012AA	1C 1A		mov	r2,r3	
ok	ST:000012AC	1C 3B		mov	r3,r7	
ok	ST:000012AE	33 22		add	r3,#0x22	
ok	ST:000012B0	80 1A		strh	r2,[r3]	
partial	702		*sign = ((mcount % period) >= period/2) ? -1 : +1;*			
ok	ST:000012B2	4B B2		ldr	r3,0x157C	
ok	ST:000012B4	68 1A		ldr	r2,[r3]	
ok	ST:000012B6	4B B2		ldr	r3,0x1580	
ok	ST:000012B8	68 1B		ldr	r3,[r3]	
ok	ST:000012BA	1C 10		mov	r0,r2	
ok	ST:000012BC	1C 19		mov	r1,r3	
ok	ST:000012BE	F0 03 F8 37		bl	0x4330	
ok	ST:000012C2	1C 0B		mov	r3,r1	
ok	ST:000012C4	1C 1A		mov	r2,r3	
ok	ST:000012C6	4B AE		ldr	r3,0x1580	
ok	ST:000012C8	68 1B		ldr	r3,[r3]	
ok	ST:000012CA	2B 00		cmp	r3,#0x0	
taken	ST:000012CC	DA 00		bge	0x12D0	
never	ST:000012CE	33 01		add	r3,#0x1	
ok	ST:000012D0	10 5B		asr	r3,r3,#0x1	
ok	ST:000012D2	42 9A		cmp	r2,r3	
taken	ST:000012D4	DB 01		blt	0x12DA	
never	ST:000012D6	4B AC		ldr	r3,0x1588	
never	ST:000012D8	E0 00		b	0x12DC	
ok	ST:000012DA	23 01		mov	r3,#0x1	
ok	ST:000012DC	1C 3A		mov	r2,r7	
ok	ST:000012DE	32 20		add	r2,#0x20	
ok	ST:000012E0	80 13		strh	r3,[r2]	
ok	703		*plot1 = plot1 + sign * inc;*			
ok	ST:000012E2	1C 3B		mov	r3,r7	
ok	ST:000012E4	33 20		add	r3,#0x20	
ok	ST:000012E6	88 1B		ldrh	r3,[r3]	
ok	ST:000012E8	1C 3A		mov	r2,r7	
ok	ST:000012EA	32 22		add	r2,#0x22	
ok	ST:000012EC	88 12		ldrh	r2,[r2]	
ok	ST:000012EE	43 53		mul	r3,r2	
ok	ST:000012F0	04 1B		lsl	r3,r3,#0x10	
ok	ST:000012F2	0C 1A		lsr	r2,r3,#0x10	
ok	ST:000012F4	4B A5		ldr	r3,0x158C	
ok	ST:000012F6	88 1B		ldrh	r3,[r3]	
ok	ST:000012F8	04 1B		lsl	r3,r3,#0x10	
ok	ST:000012FA	0C 1B		lsr	r3,r3,#0x10	
ok	ST:000012FC	18 D3		add	r3,r2,r3	
ok	ST:000012FE	04 1B		lsl	r3,r3,#0x10	
ok	ST:00001300	0C 1B		lsr	r3,r3,#0x10	
ok	ST:00001302	04 1B		lsl	r3,r3,#0x10	
ok	ST:00001304	0C 1A		lsr	r2,r3,#0x10	
ok	ST:00001306	4B A1		ldr	r3,0x158C	
ok	ST:00001308	80 1A		strh	r2,[r3]	
ok	704		*plot2 = 25000 * sign;*			
ok	ST:0000130A	1C 3B		mov	r3,r7	
ok	ST:0000130C	33 20		add	r3,#0x20	
ok	ST:0000130E	88 1B		ldrh	r3,[r3]	
ok	ST:00001310	4A 9F		ldr	r2,0x1590	
ok	ST:00001312	43 53		mul	r3,r2	
ok	ST:00001314	04 1B		lsl	r3,r3,#0x10	
ok	ST:00001316	0C 1B		lsr	r3,r3,#0x10	
ok	ST:00001318	04 1B		lsl	r3,r3,#0x10	
ok	ST:0000131A	0C 1A		lsr	r2,r3,#0x10	
ok	ST:0000131C	4B 9D		ldr	r3,0x1594	
ok	ST:0000131E	80 1A		strh	r2,[r3]	
	705					
ok	706		*vtriplearray[0][0][0] = 1;*			
ok	ST:00001320	4B 9D		ldr	r3,0x1598	
ok	ST:00001322	22 01		mov	r2,#0x1	
ok	ST:00001324	70 1A		strb	r2,[r3]	
ok	707		*vtriplearray[1][0][0] = 2;*			
ok	ST:00001326	4B 9C		ldr	r3,0x1598	

Coverage	Addr/Line	Code	Label	Mnemonics		Comment/Justification
ok	ST:00001328	22 02		mov	r2,#0x2	
ok	ST:0000132A	73 1A		strb	r2,[r3,#0x0C]	
ok	708		*vtriplearray[0][1][0] = 3;*			
ok	ST:0000132C	4B 9A		ldr	r3,0x1598	
ok	ST:0000132E	22 03		mov	r2,#0x3	
ok	ST:00001330	71 1A		strb	r2,[r3,#0x4]	
ok	709		*vtriplearray[0][0][1] = 4;*			
ok	ST:00001332	4B 99		ldr	r3,0x1598	
ok	ST:00001334	22 04		mov	r2,#0x4	
ok	ST:00001336	70 5A		strb	r2,[r3,#0x1]	
	710					
ok	711		*func2();*			
ok	ST:00001338	F7 FF F8 CE		bl	0x4D8	
ok	712		*func2a();*			
ok	ST:0000133C	F7 FF F9 12		bl	0x564	
ok	713		*func2b();*			
ok	ST:00001340	F7 FF F9 3E		bl	0x5C0	
ok	714		*func2c();*			
ok	ST:00001344	F7 FF F9 5C		bl	0x600	
ok	715		*func2d();*			
ok	ST:00001348	F7 FF F9 BA		bl	0x6C0	
	716					
ok	717		*funcptr = (int (*) ()) 0;*			
ok	ST:0000134C	4B 93		ldr	r3,0x159C	
ok	ST:0000134E	22 00		mov	r2,#0x0	
ok	ST:00001350	60 1A		str	r2,[r3]	
ok	718		*funcptr = func3;*			
ok	ST:00001352	4B 92		ldr	r3,0x159C	
ok	ST:00001354	4A 92		ldr	r2,0x15A0	
ok	ST:00001356	60 1A		str	r2,[r3]	
	719					
ok	720		*init_linked_list();*			
ok	ST:00001358	F7 FF FE 70		bl	0x103C	
ok	721		*ast.count = 12345;*			
ok	ST:0000135C	4B 91		ldr	r3,0x15A4	
ok	ST:0000135E	4A 92		ldr	r2,0x15A8	
ok	ST:00001360	60 5A		str	r2,[r3,#0x4]	
ok	722		*ast.field1 = 1 + mstatic2;*			
ok	ST:00001362	4B 85		ldr	r3,0x1578	
ok	ST:00001364	68 1B		ldr	r3,[r3]	
ok	ST:00001366	06 1B		lsl	r3,r3,#0x18	
ok	ST:00001368	0E 1B		lsr	r3,r3,#0x18	
ok	ST:0000136A	33 01		add	r3,#0x1	
ok	ST:0000136C	06 1B		lsl	r3,r3,#0x18	
ok	ST:0000136E	0E 1B		lsr	r3,r3,#0x18	
ok	ST:00001370	06 1B		lsl	r3,r3,#0x18	
ok	ST:00001372	0E 1B		lsr	r3,r3,#0x18	
ok	ST:00001374	01 9B		lsl	r3,r3,#0x6	
ok	ST:00001376	06 1B		lsl	r3,r3,#0x18	
ok	ST:00001378	16 1B		asr	r3,r3,#0x18	
ok	ST:0000137A	11 9B		asr	r3,r3,#0x6	
ok	ST:0000137C	06 1B		lsl	r3,r3,#0x18	
ok	ST:0000137E	0E 19		lsr	r1,r3,#0x18	
ok	ST:00001380	4B 88		ldr	r3,0x15A4	
ok	ST:00001382	22 03		mov	r2,#0x3	
ok	ST:00001384	40 0A		and	r2,r1	
ok	ST:00001386	7C 19		ldrb	r1,[r3,#0x10]	
ok	ST:00001388	20 03		mov	r0,#0x3	
ok	ST:0000138A	43 81		bic	r1,r0	
ok	ST:0000138C	43 0A		orr	r2,r1	
ok	ST:0000138E	74 1A		strb	r2,[r3,#0x10]	
ok	723		*ast.field2 = 2;*			
ok	ST:00001390	4B 84		ldr	r3,0x15A4	
ok	ST:00001392	7C 1A		ldrb	r2,[r3,#0x10]	
ok	ST:00001394	21 1C		mov	r1,#0x1C	
ok	ST:00001396	43 8A		bic	r2,r1	
ok	ST:00001398	21 08		mov	r1,#0x8	
ok	ST:0000139A	43 0A		orr	r2,r1	

Coverage	Addr/Line	Code	Label	Mnemonic		Comment/Justification
	724					
ok	725		*ast = func4(ast);*			
ok	ST:0000139E	4C 81		ldr	r4,0x15A4	
ok	ST:000013A0	1C 38		mov	r0,r7	
ok	ST:000013A2	4B 80		ldr	r3,0x15A4	
ok	ST:000013A4	46 6A		mov	r2,r13	
ok	ST:000013A6	1C 19		mov	r1,r3	
ok	ST:000013A8	31 0C		add	r1,#0x0C	
ok	ST:000013AA	C9 60		ldmia	r1!,{r5-r6}	
ok	ST:000013AC	C2 60		stmia	r2!,{r5-r6}	
ok	ST:000013AE	68 19		ldr	r1,[r3]	
ok	ST:000013B0	68 5A		ldr	r2,[r3,#0x4]	
ok	ST:000013B2	68 9B		ldr	r3,[r3,#0x8]	
ok	ST:000013B4	F7 FF F9 BA		bl	0x72C	
ok	ST:000013B8	1C 3A		mov	r2,r7	
ok	ST:000013BA	1C 23		mov	r3,r4	
ok	ST:000013BC	CA 13		ldmia	r2!,{r0-r1,r4}	
ok	ST:000013BE	C3 13		stmia	r3!,{r0-r1,r4}	
ok	ST:000013C0	CA 60		ldmia	r2!,{r5-r6}	
ok	ST:000013C2	C3 60		stmia	r3!,{r5-r6}	
	726					
ok	727		*j = (*funcptr) ();*			
ok	ST:000013C4	4B 75		ldr	r3,0x159C	
ok	ST:000013C6	68 1B		ldr	r3,[r3]	
ok	ST:000013C8	F0 00 F9 58		bl	0x167C	
ok	ST:000013CC	1C 03		mov	r3,r0	
ok	ST:000013CE	62 7B		str	r3,[r7,#0x24]	
ok	ST:000013D0	E0 00		b	0x13D4	
never	733		*goto start;*			
never	ST:000013D2	46 C0		nop		j is never 0, seems to be useless code
	728					
	729		*start:*			
ok	730		*j = func5((int) j, (char) 2, (long) 3);*			
ok	ST:000013D4	6A 7B	start:	ldr	r3,[r7,#0x24]	
ok	ST:000013D6	1C 18		mov	r0,r3	
ok	ST:000013D8	21 02		mov	r1,#0x2	
ok	ST:000013DA	22 03		mov	r2,#0x3	
ok	ST:000013DC	F7 FF F9 CE		bl	0x77C	
ok	ST:000013E0	1C 03		mov	r3,r0	
ok	ST:000013E2	62 7B		str	r3,[r7,#0x24]	
	731					
not taken	732		*if (j == 0)*			
ok	ST:000013E4	6A 7B		ldr	r3,[r7,#0x24]	
ok	ST:000013E6	2B 00		cmp	r3,#0x0	
not taken	ST:000013E8	D0 F3		beq	0x13D2	
	734					
ok	735		*enumvar = enumx;*			
ok	ST:000013EA	4B 70		ldr	r3,0x15AC	
ok	ST:000013EC	22 FF		mov	r2,#0xFF	
ok	ST:000013EE	70 1A		strb	r2,[r3]	
	736					
ok	737		*vfloat = 2.0;*			
ok	ST:000013F0	4B 6F		ldr	r3,0x15B0	
ok	ST:000013F2	4A 70		ldr	r2,0x15B4	
ok	ST:000013F4	60 1A		str	r2,[r3]	
ok	738		*func6(vfloat, (float) 3.0);*			
ok	ST:000013F6	4B 6E		ldr	r3,0x15B0	
ok	ST:000013F8	68 1A		ldr	r2,[r3]	
ok	ST:000013FA	4B 6F		ldr	r3,0x15B8	
ok	ST:000013FC	1C 10		mov	r0,r2	
ok	ST:000013FE	1C 19		mov	r1,r3	
ok	ST:00001400	F7 FF F9 D2		bl	0x7A8	
ok	739		*vdouble = 2.0;*			
ok	ST:00001404	4A 6D		ldr	r2,0x15BC	
ok	ST:00001406	4C 57		ldr	r4,0x1564	
ok	ST:00001408	4B 55		ldr	r3,0x1560	

Coverage	Addr/Line	Code	Label	Mnemonics	Comment/Justification
ok	ST:0000140A	60 13		str r3,[r2]	
ok	ST:0000140C	60 54		str r4,[r2,#0x4]	
ok	740	*func7(vdouble, (double) 3.0);*			
ok	ST:0000140E	4B 6B		ldr r3,0x15BC	
ok	ST:00001410	68 19		ldr r1,[r3]	
ok	ST:00001412	68 5A		ldr r2,[r3,#0x4]	
ok	ST:00001414	4B 54		ldr r3,0x1568	
ok	ST:00001416	4C 55		ldr r4,0x156C	
ok	ST:00001418	1C 08		mov r0,r1	
ok	ST:0000141A	1C 11		mov r1,r2	
ok	ST:0000141C	1C 1A		mov r2,r3	
ok	ST:0000141E	1C 23		mov r3,r4	
ok	ST:00001420	F7 FF F9 E8		bl 0x7F4	
	741				
ok	742	*func8();*			
ok	ST:00001424	F7 FF FA 22		bl 0x86C	
ok	743	*func9();*			
ok	ST:00001428	F7 FF FB 00		bl 0xA2C	
ok	744	*func10();*			
ok	ST:0000142C	F7 FF FB 2C		bl 0xA88	
ok	745	*func11(5);*			
ok	ST:00001430	20 05		mov r0,#0x5	
ok	ST:00001432	F7 FF FB F9		bl 0xC28	
ok	746	*func13(1, 2, 3);*			
ok	ST:00001436	20 01		mov r0,#0x1	
ok	ST:00001438	21 02		mov r1,#0x2	
ok	ST:0000143A	22 03		mov r2,#0x3	
ok	ST:0000143C	F7 FF FC 36		bl 0xCAC	
ok	747	*func14((char) 55);*			
ok	ST:00001440	20 37		mov r0,#0x37	
ok	ST:00001442	F7 FF FC 63		bl 0xD0C	
ok	748	*func15((short) 55);*			
ok	ST:00001446	20 37		mov r0,#0x37	
ok	ST:00001448	F7 FF FC 70		bl 0xD2C	
ok	749	*func16((long) 55);*			
ok	ST:0000144C	20 37		mov r0,#0x37	
ok	ST:0000144E	F7 FF FC 7D		bl 0xD4C	
ok	750	*func17((short) 44, (short) 55);*			
ok	ST:00001452	20 2C		mov r0,#0x2C	
ok	ST:00001454	21 37		mov r1,#0x37	
ok	ST:00001456	F7 FF FC 85		bl 0xD64	
ok	751	*func18((short) 44, (long) 55);*			
ok	ST:0000145A	20 2C		mov r0,#0x2C	
ok	ST:0000145C	21 37		mov r1,#0x37	
ok	ST:0000145E	F7 FF FC 97		bl 0xD90	
ok	752	*func19((long) 44, (short) 55);*			
ok	ST:00001462	20 2C		mov r0,#0x2C	
ok	ST:00001464	21 37		mov r1,#0x37	
ok	ST:00001466	F7 FF FC A5		bl 0xDB4	
ok	753	*func20((short) 33, (short) 44, (short) 55);*			
ok	ST:0000146A	20 21		mov r0,#0x21	
ok	ST:0000146C	21 2C		mov r1,#0x2C	
ok	ST:0000146E	22 37		mov r2,#0x37	
ok	ST:00001470	F7 FF FC B2		bl 0xDD8	
ok	754	*func21((long) 33, (short) 44, (short) 55);*			
ok	ST:00001474	20 21		mov r0,#0x21	
ok	ST:00001476	21 2C		mov r1,#0x2C	
ok	ST:00001478	22 37		mov r2,#0x37	
ok	ST:0000147A	F7 FF FC C7		bl 0xE0C	
ok	755	*func22((short) 33, (long) 44, (short) 55);*			
ok	ST:0000147E	20 21		mov r0,#0x21	
ok	ST:00001480	21 2C		mov r1,#0x2C	
ok	ST:00001482	22 37		mov r2,#0x37	
ok	ST:00001484	F7 FF FC DA		bl 0xE3C	
ok	756	*func23((short) 33, (short) 44, (long) 55);*			
ok	ST:00001488	20 21		mov r0,#0x21	
ok	ST:0000148A	21 2C		mov r1,#0x2C	

Coverage	Address	Code	Label	Mnemonic	Comment/Justification
OK	ST:0000148C	23 37		mov	r3,#0x37
OK	ST:0000148E	F7 FF FC ED		bl	0xE6C
	757				
never	758		*j = func24();*		
never	ST:00001492	F7 FF FD 05		bl	0xEA0
never	ST:00001496	1C 03		mov	r3,r0
never	ST:00001498	62 7B		str	r3,[r7,#0x24]
never	759		*vlong = func25();*		
never	ST:0000149A	F7 FF FD 09		bl	0xEB0
never	ST:0000149E	1C 02		mov	r2,r0
never	ST:000014A0	4B 47		ldr	r3,0x15C0
never	ST:000014A2	60 1A		str	r2,[r3]
never	760		*p = func26();*		
never	ST:000014A4	F7 FF FD 0E		bl	0xEC4
never	ST:000014A8	1C 03		mov	r3,r0
never	ST:000014AA	61 FB		str	r3,[r7,#0x1C]
never	761		*vshortrecord = func28(vshortrecord);*		
never	ST:000014AC	4B 45		ldr	r3,0x15C4
never	ST:000014AE	68 18		ldr	r0,[r3]
never	ST:000014B0	F7 FF FE AC		bl	0x120C
never	ST:000014B4	1C 03		mov	r3,r0
never	ST:000014B6	1C 3A		mov	r2,r7
never	ST:000014B8	1C 19		mov	r1,r3
never	ST:000014BA	80 11		strh	r1,[r2]
never	ST:000014BC	0C 1A		lsr	r2,r3,#0x10
never	ST:000014BE	1C BB		add	r3,r7,#0x2
never	ST:000014C0	80 1A		strh	r2,[r3]
never	ST:000014C2	4B 40		ldr	r3,0x15C4
never	ST:000014C4	1C 3A		mov	r2,r7
never	ST:000014C6	68 12		ldr	r2,[r2]
never	ST:000014C8	60 1A		str	r2,[r3]
	762				
never	763		*encode(p);*		
never	ST:000014CA	69 FB		ldr	r3,[r7,#0x1C]
never	ST:000014CC	1C 18		mov	r0,r3
never	ST:000014CE	F7 FF FE 4B		bl	0x1168
never	764		*sieve();*		
never	ST:000014D2	F0 00 F8 85		bl	0x15E0
	765				
never	766		*mstatic1 = test_cond_instr(14, 14); /* returns 0 */*		
never	ST:000014D6	20 0E		mov	r0,#0x0E
never	ST:000014D8	21 0E		mov	r1,#0x0E
never	ST:000014DA	F7 FF FD 1D		bl	0xF18
never	ST:000014DE	1C 02		mov	r2,r0
never	ST:000014E0	4B 24		ldr	r3,0x1574
never	ST:000014E2	60 1A		str	r2,[r3]
never	767		*mstatic2 = test_cond_instr(78, 55); /* returns 918 */*		
never	ST:000014E4	20 4E		mov	r0,#0x4E
never	ST:000014E6	21 37		mov	r1,#0x37
never	ST:000014E8	F7 FF FD 16		bl	0xF18
never	ST:000014EC	1C 02		mov	r2,r0
never	ST:000014EE	4B 22		ldr	r3,0x1578
never	ST:000014F0	60 1A		str	r2,[r3]
	768				
never	769		*datas.b[0] = 0x12;*		
never	ST:000014F2	4B 35		ldr	r3,0x15C8
never	ST:000014F4	22 12		mov	r2,#0x12
never	ST:000014F6	70 1A		strb	r2,[r3]
never	770		*datas.b[1] = 0x12;*		
never	ST:000014F8	4B 33		ldr	r3,0x15C8
never	ST:000014FA	22 12		mov	r2,#0x12
never	ST:000014FC	70 5A		strb	r2,[r3,#0x1]
never	771		*datas.b[2] = 0x12;*		
never	ST:000014FE	4B 32		ldr	r3,0x15C8
never	ST:00001500	22 12		mov	r2,#0x12
never	ST:00001502	70 9A		strb	r2,[r3,#0x2]
never	772		*datas.b[3] = p[0];*		
never	ST:00001504	69 FB		ldr	r3,[r7,#0x1C]

Coverage	Addr/Line	Code	Label	Mnemonics	Comment/Justification
never	ST:00001506	68 1A		ldr r2,[r3]	
never	ST:00001508	4B 2F		ldr r3,0x15C8	
never	ST:0000150A	70 DA		strb r2,[r3,#0x3]	
never	773		*datas.w[0] = 0x1234;*		
never	ST:0000150C	4B 2E		ldr r3,0x15C8	
never	ST:0000150E	4A 2F		ldr r2,0x15CC	
never	ST:00001510	80 1A		strh r2,[r3]	
never	774		*datas.w[1] = 0x1234;*		
never	ST:00001512	4B 2D		ldr r3,0x15C8	
never	ST:00001514	4A 2D		ldr r2,0x15CC	
never	ST:00001516	80 5A		strh r2,[r3,#0x2]	
never	775		*datas.l[0] = 0x12345678;*		
never	ST:00001518	4B 2B		ldr r3,0x15C8	
never	ST:0000151A	4A 2D		ldr r2,0x15D0	
never	ST:0000151C	60 1A		str r2,[r3]	
never	776		*datas.b[0] = 0x11;*		
never	ST:0000151E	4B 2A		ldr r3,0x15C8	
never	ST:00001520	22 11		mov r2,#0x11	
never	ST:00001522	70 1A		strb r2,[r3]	
never	777		*datas.b[1] = 0x11;*		
never	ST:00001524	4B 28		ldr r3,0x15C8	
never	ST:00001526	22 11		mov r2,#0x11	
never	ST:00001528	70 5A		strb r2,[r3,#0x1]	
never	778		*datas.b[2] = 0x11;*		
never	ST:0000152A	4B 27		ldr r3,0x15C8	
never	ST:0000152C	22 11		mov r2,#0x11	
never	ST:0000152E	70 9A		strb r2,[r3,#0x2]	
never	779		*datas.b[3] = 0x11;*		
never	ST:00001530	4B 25		ldr r3,0x15C8	
never	ST:00001532	22 11		mov r2,#0x11	
never	ST:00001534	70 DA		strb r2,[r3,#0x3]	
never	780		*datas.w[0] = 0x1122;*		
never	ST:00001536	4B 24		ldr r3,0x15C8	
never	ST:00001538	4A 26		ldr r2,0x15D4	
never	ST:0000153A	80 1A		strh r2,[r3]	
never	781		*datas.w[1] = 0x1122;*		
never	ST:0000153C	4B 22		ldr r3,0x15C8	
never	ST:0000153E	4A 25		ldr r2,0x15D4	
never	ST:00001540	80 5A		strh r2,[r3,#0x2]	
never	782		*datas.l[0] = 0x11223344;*		
never	ST:00001542	4B 21		ldr r3,0x15C8	
never	ST:00001544	4A 24		ldr r2,0x15D8	
never	ST:00001546	60 1A		str r2,[r3]	
	783				
never	784		*} while (vint);*		
never	ST:00001548	4B 24		ldr r3,0x15DC	
never	ST:0000154A	68 1B		ldr r3,[r3]	
never	ST:0000154C	2B 00		cmp r3,#0x0	
never	ST:0000154E	D0 00		beq 0x1552	
never	ST:00001550	E6 91		b 0x1276	
	785				
never	786		*return 0;*		
never	ST:00001552	23 00		mov r3,#0x0	
p-read	787		*}*		
never	ST:00001554	1C 18		mov r0,r3	
never	ST:00001556	46 BD		mov r13,r7	
never	ST:00001558	B0 0B		add sp,#0x2C	
never	ST:0000155A	BC F0		pop {r4-r7}	
never	ST:0000155C	BC 02		pop {r1}	
never	ST:0000155E	47 08		bx r1	
read	SP:00001560	00 00 00 00		dcd 0x0	
read	SP:00001564	40 00 00 00		dcd 0x40000000	
read	SP:00001568	00 00 00 00		dcd 0x0	
read	SP:0000156C	40 08 00 00		dcd 0x40080000	
read	SP:00001570	00 00 46 4C		dcd 0x464C	
read	SP:00001574	00 00 46 C0		dcd 0x46C0	
read	SP:00001578	00 00 46 C4		dcd 0x46C4	
read	SP:0000157C	00 00 46 C8		dcd 0x46C8	

Coverage	Addr/Line	Code	Label	Mnemonics		Comment/Justification
read	SP:00001580	00 00 46 28		dcd	0x4628	
read	SP:00001584	00 00 EA 60		dcd	0xEA60	
never	SP:00001588	00 00 FF FF		dcd	0xFFFF	
read	SP:0000158C	00 00 46 2C		dcd	0x462C	
read	SP:00001590	00 00 61 A8		dcd	0x61A8	
read	SP:00001594	00 00 46 CC		dcd	0x46CC	
read	SP:00001598	00 00 5E D0		dcd	0x5ED0	
read	SP:0000159C	00 00 49 E0		dcd	0x49E0	
read	SP:000015A0	00 00 07 1D		dcd	0x71D	
read	SP:000015A4	00 00 47 50		dcd	0x4750	
read	SP:000015A8	00 00 30 39		dcd	0x3039	
read	SP:000015AC	00 00 5D 94		dcd	0x5D94	
read	SP:000015B0	00 00 46 30		dcd	0x4630	
read	SP:000015B4	40 00 00 00		dcd	0x40000000	
read	SP:000015B8	40 40 00 00		dcd	0x40400000	
read	SP:000015BC	00 00 46 38		dcd	0x4638	
never	SP:000015C0	00 00 5E CC		dcd	0x5ECC	
never	SP:000015C4	00 00 47 C8		dcd	0x47C8	
never	SP:000015C8	00 00 47 CC		dcd	0x47CC	
never	SP:000015CC	00 00 12 34		dcd	0x1234	
never	SP:000015D0	12 34 56 78		dcd	0x12345678	
never	SP:000015D4	00 00 11 22		dcd	0x1122	
never	SP:000015D8	11 22 33 44		dcd	0x11223344	
never	SP:000015DC	00 00 46 40		dcd	0x4640	
	788					
	789					
	790	*#define SIZE 18*				
	791	*char flags[SIZE+1];*				
	792					
	793	*static int sieve(void) /* sieve of erathostenes */*				
never	794	*{*				
never	ST:000015E0	B5 F0	sieve:	push	{r4-r7,r14}	
never	ST:000015E2	B0 83		sub	sp,#0x0C	
never	ST:000015E4	AF 00		add	r7,sp,#0x0	
	795	*register int i, prime, k;*				
	796	*int count;*				
	797					
never	798	*count = 0;*				
never	ST:000015E6	23 00		mov	r3,#0x0	
never	ST:000015E8	60 7B		str	r3,[r7,#0x4]	
	799					
never	800	*for (i = 0 ; i <= SIZE ; flags[i++] = TRUE) ;*				
never	ST:000015EA	24 00		mov	r4,#0x0	
never	ST:000015EC	E0 03		b	0x15F6	
never	ST:000015EE	4B 11		ldr	r3,0x1634	
never	ST:000015F0	22 01		mov	r2,#0x1	
never	ST:000015F2	55 1A		strb	r2,[r3,r4]	
never	ST:000015F4	34 01		add	r4,#0x1	
never	ST:000015F6	2C 12		cmp	r4,#0x12	
never	ST:000015F8	DD F9		ble	0x15EE	
	801					
never	802	*for (i = 0; i <= SIZE; i++) {*				
never	ST:000015FA	24 00		mov	r4,#0x0	
never	ST:000015FC	E0 11		b	0x1622	
never	803	*if (flags[i]) {*				
never	ST:000015FE	4B 0D		ldr	r3,0x1634	
never	ST:00001600	5D 1B		ldrb	r3,[r3,r4]	
never	ST:00001602	2B 00		cmp	r3,#0x0	
never	ST:00001604	D0 0C		beq	0x1620	
never	804	*prime = i + i + 3;*				
never	ST:00001606	19 23		add	r3,r4,r4	
never	ST:00001608	1C DE		add	r6,r3,#0x3	
never	805	*k = i + prime;*				
never	ST:0000160A	19 A5		add	r5,r4,r6	
never	806	*while (k <= SIZE) {*				
never	ST:0000160C	E0 03		b	0x1616	
never	807	*flags[k] = FALSE;*				
never	ST:0000160E	4B 09		ldr	r3,0x1634	
	ST:00001610			mov	r2,#0x0	

Coverage	Address	Code	Label	Mnemonic	Operands	Comment/Justification
never	ST:00001612	66 80		ldrb	r3,[r3,r5]	
never	808		*k += prime;*			
never	ST:00001614	19 AD		add	r5,r5,r6	
never	806		*while (k <= SIZE) {*			
never	ST:00001616	2D 12		cmp	r5,#0x12	
never	ST:00001618	DD F9		ble	0x160E	
	809		*}*			
never	810		*count++;*			
never	ST:0000161A	68 7B		ldr	r3,[r7,#0x4]	
never	ST:0000161C	33 01		add	r3,#0x1	
never	ST:0000161E	60 7B		str	r3,[r7,#0x4]	
	801					
never	802		*for (i = 0; i <= SIZE; i++) {*			
never	ST:00001620	34 01		add	r4,#0x1	
never	ST:00001622	2C 12		cmp	r4,#0x12	
never	ST:00001624	DD EB		ble	0x15FE	
	811		*}*			
	812		*}*			
	813					
never	814		*return count;*			
never	ST:00001626	68 7B		ldr	r3,[r7,#0x4]	
never	815		*}*			
never	ST:00001628	1C 18		mov	r0,r3	
never	ST:0000162A	46 BD		mov	r13,r7	
never	ST:0000162C	B0 03		add	sp,#0x0C	
never	ST:0000162E	BC F0		pop	{r4-r7}	
never	ST:00001630	BC 02		pop	{r1}	
never	ST:00001632	47 08		bx	r1	
never	SP:00001634	00 00 5D 98		dcd	0x5D98	
	816					
	817					
	818		*int background(void) /* job for background-demo */*			
never	819		*{*			
never	ST:00001638	B5 80	**background:**	push	{r7,r14}	never called => dead code
never	ST:0000163A	AF 00		add	r7,sp,#0x0	
	820		*static long int bcnt1, bcnt2;*			
	821					
never	822		*bcnt1 = bcnt2 = 0;*			
never	ST:0000163C	4B 0C		ldr	r3,0x1670	
never	ST:0000163E	22 00		mov	r2,#0x0	
never	ST:00001640	60 1A		str	r2,[r3]	
never	ST:00001642	4B 0B		ldr	r3,0x1670	
never	ST:00001644	68 1A		ldr	r2,[r3]	
never	ST:00001646	4B 0B		ldr	r3,0x1674	
never	ST:00001648	60 1A		str	r2,[r3]	
	823					
	824		*while (1) {*			
never	825		*bcnt1 = 10000000;*			
never	ST:0000164A	4B 0A		ldr	r3,0x1674	
never	ST:0000164C	4A 0A		ldr	r2,0x1678	
never	ST:0000164E	60 1A		str	r2,[r3]	
never	826		*while (bcnt1>0)*			
never	ST:00001650	E0 04		b	0x165C	
never	827		*bcnt1--;*			
never	ST:00001652	4B 08		ldr	r3,0x1674	
never	ST:00001654	68 1B		ldr	r3,[r3]	
never	ST:00001656	1E 5A		sub	r2,r3,#0x1	
never	ST:00001658	4B 06		ldr	r3,0x1674	
never	ST:0000165A	60 1A		str	r2,[r3]	
never	826		*while (bcnt1>0)*			
never	ST:0000165C	4B 05		ldr	r3,0x1674	
never	ST:0000165E	68 1B		ldr	r3,[r3]	
never	ST:00001660	2B 00		cmp	r3,#0x0	
never	ST:00001662	DC F6		bgt	0x1652	
never	828		*bcnt2++;*			
never	ST:00001664	4B 02		ldr	r3,0x1670	
never	ST:00001666	68 1B		ldr	r3,[r3]	
never	ST:00001668	1C 5A		add	r2,r3,#0x1	

Coverage	Addr/Line	Code	Label	Mnemonics	Comment/Justification
never	ST:0000166A	4B 01		ldr r3,0x1670	
never	ST:0000166C	60 1A		str r2,[r3]	
partial	829	}			
never	ST:0000166E	E7 EC		b 0x164A	
never	SP:00001670	00 00 46 D0		dcd 0x46D0	
never	SP:00001674	00 00 46 D4		dcd 0x46D4	
never	SP:00001678	00 98 96 80		dcd 0x989680	
ok	ST:0000167C	00 18		movs r0,r3	