

**Source Code**

```
int main() {
```

Offset	Bytecode	Assembly Instruction	Set Fault	Reset Fault	Flip Fault
0x34	b590	push {r4, r7, lr}	0	0	0
0x36	b08d	sub sp, #52 ; 0x34	0	0	0
0x38	af00	add r7, sp, #0	0	0	0
0x3a	4a36	ldr r2, [pc, #216] ; (e0 <main+0xe0>)	0	0	0
0x3c	447a	add r2, pc	0	0	0
0x3e	4b36	ldr r3, [pc, #216] ; (e4 <main+0xe4>)	0	0	0
0x40	58d3	ldr r3, [r2, r3]	0	0	0
0x42	681b	ldr r3, [r3, #0]	0	0	0
0x44	62fb	str r3, [r7, #44] ; 0x2c	0	0	0
0x46	f04f 0300	mov.w r3, #0	0	0	0

**Source Code**

```
int PINSize = 4;
```

Offset	Bytecode	Assembly Instruction	Set Fault	Reset Fault	Flip Fault
0x4a	2304	movs r3, #4	33.33	100.00	22.22
0x4c	60bb	str r3, [r7, #8]	11.11	77.78	66.67

**Source Code**

```
int PINCandidate[] = {0,0,0,0};
```

Offset	Bytecode	Assembly Instruction	Set Fault	Reset Fault	Flip Fault
0x4e	f107 030c	add.w r3, r7, #12	0	24.0	0
0x52	2200	movs r2, #0	0	0.0	0
0x54	601a	str r2, [r3, #0]	0	0.0	0
0x56	605a	str r2, [r3, #4]	0	0.0	0
0x58	609a	str r2, [r3, #8]	0	0.0	0
0x5a	60da	str r2, [r3, #12]	0	0.0	0

**Source Code**

```
int PINTrue[] = {1,2,3,4};
```

Offset	Bytecode	Assembly Instruction	Set Fault	Reset Fault	Flip Fault
0x5c	4b2f	ldr r3, [pc, #188] ; (e8 <main+0xe8>)	0.0	0	0.0
0x5e	447b	add r3, pc	0.0	0	0.0
0x60	f107 041c	add.w r4, r7, #28	0.0	0	0.0
0x64	cb0f	ldmia r3, {r0, r1, r2, r3}	0.0	0	0.0
0x66	e884 000f	stmia.w r4, {r0, r1, r2, r3}	8.0	0	0.0

**Source Code**

```
bool grantAccess = false;
```

Offset	Bytecode	Assembly Instruction	Set Fault	Reset Fault	Flip Fault
0x6a	2300	movs r3, #0	0	0	0
0x6c	70bb	strb r3, [r7, #2]	0	0	0

#### Source Code

```
bool badValue = false;
```

Offset	Bytecode	Assembly Instruction	Set Fault	Reset Fault	Flip Fault
0x6e	2300	movs r3, #0	0	0	0
0x70	70fb	strb r3, [r7, #3]	0	0	0

#### Source Code

```
int i = 0;
```

Offset	Bytecode	Assembly Instruction	Set Fault	Reset Fault	Flip Fault
0x72	2300	movs r3, #0	22.22	0	22.22
0x74	607b	str r3, [r7, #4]	0.00	0	0.00

#### Source Code

```
while (i < PINSize) {
```

Offset	Bytecode	Assembly Instruction	Set Fault	Reset Fault	Flip Fault
0x76	e012	b.n 6a <main+0x6a>	0	0	0

#### Source Code

```
if (PINCandidate[i] != PINTrue[i]) {
```

Offset	Bytecode	Assembly Instruction	Set Fault	Reset Fault	Flip Fault
0x78	687b	ldr r3, [r7, #4]	0.00	0.00	0.00
0x7a	009b	lsls r3, r3, #2	0.00	0.00	0.00
0x7c	3330	adds r3, #48 ; 0x30	0.00	0.00	0.00
0x7e	443b	add r3, r7	0.00	0.00	0.00
0x80	f853 2c24	ldr.w r2, [r3, #-36]	0.00	0.00	0.00
0x84	687b	ldr r3, [r7, #4]	0.00	0.00	0.00
0x86	009b	lsls r3, r3, #2	11.11	0.00	11.11
0x88	3330	adds r3, #48 ; 0x30	0.00	100.00	0.00
0x8a	443b	add r3, r7	0.00	11.11	0.00
0x8c	f853 3c14	ldr.w r3, [r3, #-20]	8.00	4.00	8.00
0x90	429a	cmp r2, r3	0.00	0.00	0.00
0x92	d001	beq.n 64 <main+0x64>	33.33	0.00	0.00

#### Source Code

```
badValue = true;
```

Offset	Bytecode	Assembly Instruction	Set Fault	Reset Fault	Flip Fault
0x94	2301	movs r3, #1	22.22	100.00	55.56
0x96	70fb	strb r3, [r7, #3]	11.11	77.78	77.78

#### Source Code

```
}
```

#### Source Code

```
i++;
```

Offset	Bytecode	Assembly Instruction	Set Fault	Reset Fault	Flip Fault
0x98	687b	ldr r3, [r7, #4]	0	0	0
0x9a	3301	adds r3, #1	0	0	0
0x9c	607b	str r3, [r7, #4]	0	0	0

#### Source Code

```
while (i < PINSize) {
```

Offset	Bytecode	Assembly Instruction	Set Fault	Reset Fault	Flip Fault
0x9e	687a	ldr r2, [r7, #4]	0.00	0.0	0.00
0xa0	68bb	ldr r3, [r7, #8]	33.33	0.0	77.78
0xa2	429a	cmp r2, r3	11.11	100.0	33.33
0xa4	dbe8	blt.n 44 <main+0x44>	33.33	100.0	55.56

#### Source Code

```
}
```

#### Source Code

```
if (badValue == false) {
```

Offset	Bytecode	Assembly Instruction	Set Fault	Reset Fault	Flip Fault
0xa6	78fb	ldrb r3, [r7, #3]	11.11	77.78	77.78
0xa8	f083 0301	eor.w r3, r3, #1	16.00	68.00	36.00
0xac	b2db	uxtb r3, r3	0.00	0.00	0.00
0xae	2b00	cmp r3, #0	66.67	44.44	11.11
0xb0	d001	beq.n 82 <main+0x82>	11.11	77.78	88.89

#### Source Code

```
grantAccess = true;
```

Offset	Bytecode	Assembly Instruction	Set Fault	Reset Fault	Flip Fault
0xb2	2301	movs r3, #1	0	0	0
0xb4	70bb	strb r3, [r7, #2]	0	0	0

#### Source Code

```
}
```

#### Source Code

```
if (grantAccess) {
```

Offset	Bytecode	Assembly Instruction	Set Fault	Reset Fault	Flip Fault
0xb6	78bb	ldrb r3, [r7, #2]	0	0	0
0xb8	2b00	cmp r3, #0	0	0	0
0xba	d005	beq.n 94 <main+0x94>	0	0	0

#### Source Code

```
printf("Access Granted");
```

Offset	Bytecode	Assembly Instruction	Set Fault	Reset Fault	Flip Fault
0xbc	4b18	ldr r3, [pc, #96] ; (ec <main+0xec>)	0	0	0
0xbe	447b	add r3, pc	0	0	0
0xc0	4618	mov r0, r3	0	0	0
0xc2	f7ff fffe	bl 0 <printf>	0	0	0

Offset	Bytecode	Assembly Instruction	Set Fault	Reset Fault	Flip Fault
0xc6	e004	b.n 9e <main+0x9e>	0	0	0

#### Source Code

```
} else {
```

#### Source Code

```
printf("Access Denied");
```

Offset	Bytecode	Assembly Instruction	Set Fault	Reset Fault	Flip Fault
0xc8	4b16	ldr r3, [pc, #88] ; (f0 <main+0xf0>)	0	0	0
0xca	447b	add r3, pc	0	0	0
0xcc	4618	mov r0, r3	0	0	0
0xce	f7ff fffe	bl 0 <printf>	0	0	0

#### Source Code

```
}
```

#### Source Code

```
assert(!(grantAccess == true && PINCandidate != PINTrue));
```

Offset	Bytecode	Assembly Instruction	Set Fault	Reset Fault	Flip Fault
0xd2	78bb	ldrb r3, [r7, #2]	55.56	0.0	11.11
0xd4	f083 0301	eor.w r3, r3, #1	0.00	60.0	0.00
0xd8	b2db	uxtb r3, r3	0.00	0.0	0.00
0xda	2b00	cmp r3, #0	66.67	0.0	44.44
0xdc	d108	bne.n bc <main+0xbc>	33.33	100.0	88.89
0xde	4b12	ldr r3, [pc, #72] ; (f4 <main+0xf4>)	0.00	0.0	0.00
0xe0	447b	add r3, pc	0.00	0.0	0.00
0xe2	221c	movs r2, #28	0.00	0.0	0.00
0xe4	4911	ldr r1, [pc, #68] ; (f8 <main+0xf8>)	0.00	0.0	0.00
0xe6	4479	add r1, pc	0.00	0.0	0.00
0xe8	4811	ldr r0, [pc, #68] ; (fc <main+0xfc>)	0.00	0.0	0.00
0xea	4478	add r0, pc	0.00	0.0	0.00
0xec	f7ff fffe	bl 0 <__assert_fail>	0.00	0.0	0.00

#### Source Code

```
return 0;
```

Offset	Bytecode	Assembly Instruction	Set Fault	Reset Fault	Flip Fault
0xf0	2300	movs r3, #0	0	0	0

#### Source Code

```
int main() {
```

Offset	Bytecode	Assembly Instruction	Set Fault	Reset Fault	Flip Fault
0xf2	4910	ldr r1, [pc, #64] ; (100 <main+0x100>)	0	0	0
0xf4	4479	add r1, pc	0	0	0
0xf6	4a08	ldr r2, [pc, #32] ; (e4 <main+0xe4>)	0	0	0
0xf8	588a	ldr r2, [r1, r2]	0	0	0
0xfa	6811	ldr r1, [r2, #0]	0	0	0
0xfc	6afa	ldr r2, [r7, #44] ; 0x2c	0	0	0
0xfe	4051	eors r1, r2	0	0	0
0x100	f04f 0200	mov.w r2, #0	0	0	0
0x104	d001	beq.n d6 <main+0xd6>	0	0	0
0x106	f7ff fffe	bl 0 <__stack_chk_fail>	0	0	0
0x10a	4618	mov r0, r3	0	0	0
0x10c	3734	adds r7, #52 ; 0x34	0	0	0
0x10e	46bd	mov sp, r7	0	0	0
0x110	bd90	pop {r4, r7, pc}	0	0	0
0x112	bf00	nop	0	0	0

#### Source Code

e0:\t000000d4 \t.word\t0x000000d4

#### Source Code

e4:\t00000000 \t.word\t0x00000000

#### Source Code

e8:\t0000012a \t.word\t0x0000012a

#### Source Code

ec:\t0000005e \t.word\t0x0000005e

#### Source Code

f0:\t00000066 \t.word\t0x00000066

#### Source Code

f4:\t000000c4 \t.word\t0x000000c4

#### Source Code

f8:\t00000062 \t.word\t0x00000062

#### Source Code

fc:\t0000007e \t.word\t0x0000007e

#### Source Code

100:\t0000003c \t.word\t0x0000003c