



# Spot the difference(s)

Python

```
def main():  
    print("Hello, World!")  
  
if __name__ == "__main__":  
    main()
```

C

```
int main(void) {  
    printf("Hello, World!");  
    return 0;  
}
```

# Spot the difference(s)

Python

```
def main():  
    print("Hello, World!")
```

```
if __name__ == "__main__":  
    main()
```

C

```
int main(void) {  
    printf("Hello, World!");  
    return 0;  
}
```

# Spot the difference(s)

## Python

```
def divide(a, b):  
    return a / b  
  
def main():  
    a, b = 5, 2  
    q = divide(a, b)  
    print(s)  
  
if __name__ == "__main__":  
    main()
```

## C

```
float divide (int a, int b) {  
    return a / b;  
}  
  
int main (void) {  
    int a = 5;  
    Int b = 2;  
    float q = divide(a, b);  
    printf("%f", q);  
    return 0;  
}
```

# Spot the difference(s)

## Python

```
def divide(a, b):  
    return a / b  
  
def main():  
    a, b = 5, 2  
    q = divide(a, b)  
    print(s)  
  
if __name__ == "__main__":  
    main()
```

## C

```
float divide (int a, int b) {  
    return a / b;  
}  
  
int main (void) {  
    int a = 5;  
    Int b = 2;  
    float q = divide(a, b);  
    printf("%f", q);  
    return 0;  
}
```

# Spot the difference(s)

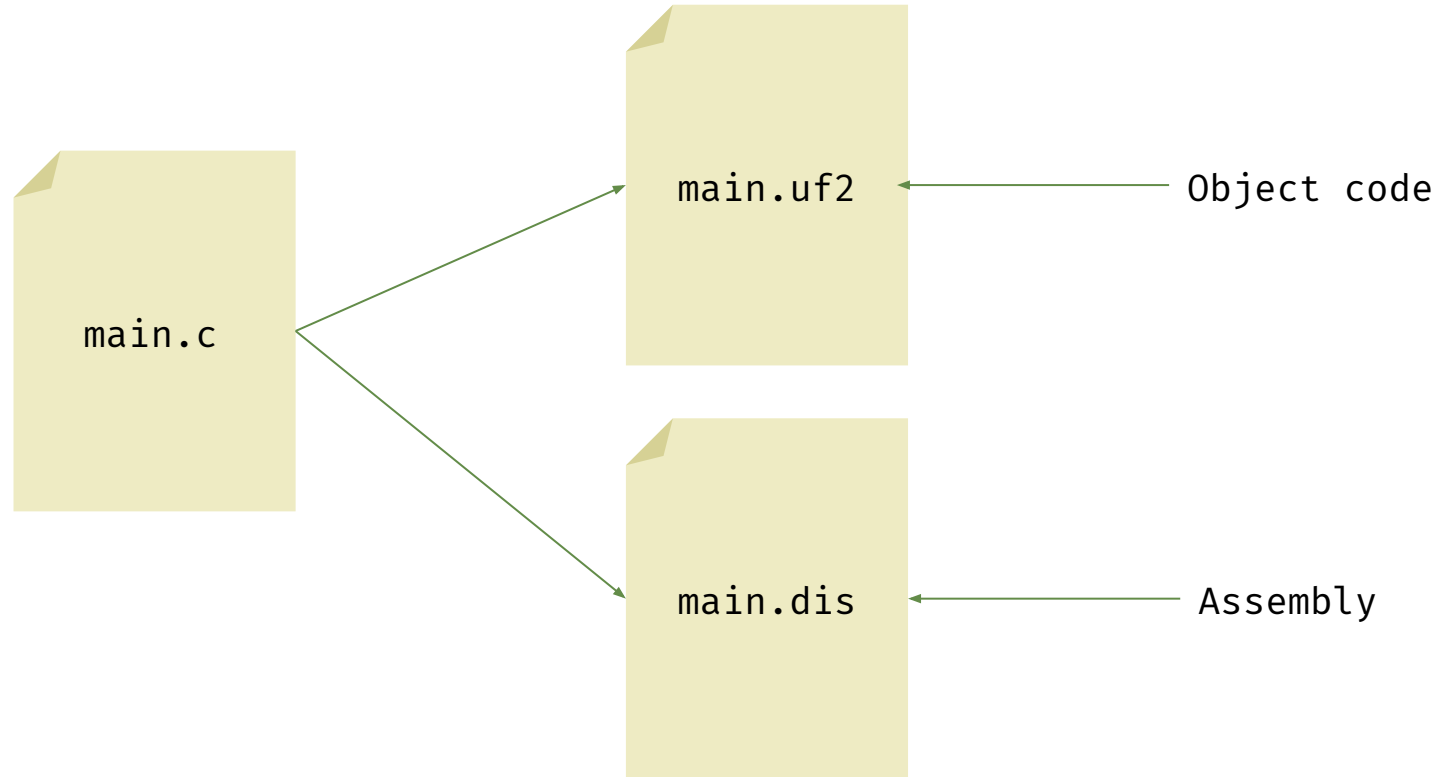
## Assembly

byte	8 bits
half word	16 bits
word	32 bits
quad	64 bits

## C

char	1 byte
short	2 bytes
int	4 bytes
float	4 bytes
long	4 bytes
long long	8 bytes
double	8 bytes

# .dis respect



# .dis respect

program.dis (hello\_world)

10000354 <main>:

10000354:	b510	push	{r4, lr}	STACK FRAME
-----------	------	------	----------	-------------

10000356:	f003 fe93	bl	10004080 <stdio_init_all>
-----------	-----------	----	---------------------------

1000035a:	4802	ldr	r0, [pc, #8]; (10000364 <main+0x10>)
-----------	------	-----	--------------------------------------

1000035c:	f003 fe54	bl	10004008 <__wrap_puts>	PRINTF
-----------	-----------	----	------------------------	--------

10000360:	2000	movs	r0, #0
-----------	------	------	--------

10000362:	bd10	pop	{r4, pc}	UNROLL STACK FRAME
-----------	------	-----	----------	--------------------

10000364:	10006990	.word	0x10006990
-----------	----------	-------	------------





# .dis respect

## program.dis (adder)

10000354 <add>:

10000354: 1840	adds r0, r0, r1	int sum = a + b;
10000356: 4770	bx lr	return sum;

10000358 <main>:

10000358: b510	push {r4, lr}	STACK FRAME
1000035a: f003 fe99	bl 10004090 <stdio_init_all>	
1000035e: 2102	movs r1, #2	int a = 2;
10000360: 2003	movs r0, #3	int b = 3;
10000362: f7ff fff7	bl 10000354 <add>	add(a, b);

# .dis respect

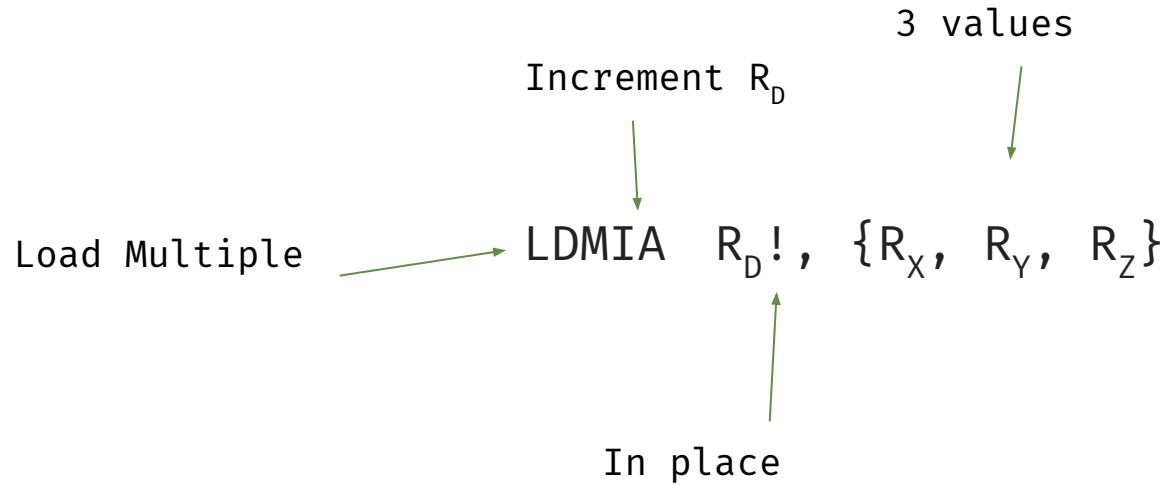
program.dis (adder; with array)

10000356: b08b	sub sp, #44 ; 0x2c	MOVE DOWN IN STACK 44 BYTES OF SPACE
10000358: f003 fea2	bl 100040a0 <stdio_init_all>	
1000035c: 4669	mov r1, sp	
1000035e: 4808	ldr r0, [pc, #32] ; (10000380 <main+0x2c>)	LOAD ARRAY $M_0$
10000360: 000b	movs r3, r1	HOLD STACK POINTER IN R3
10000362: 0002	movs r2, r0	HOLD ADDRESS OF ARRAY IN R2
10000364: ca31	ldmia r2!, {r0, r4, r5}	LOAD MULTIPLE, +4 AFTER
10000366: c331	stmia r3!, {r0, r4, r5}	STORE MULTIPLE, +4 AFTER

...

PC Program counter (CARDIAC “bug”); current spot in program

# LD...M...I...A?



# But don't take my word for it...

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS **MEMORY** XRTOS SERIAL MONITOR

0x100069b0 + ✎ 📄 🔍 🔄 ⚙️ Status: Debugger attached, stopped

	07-00	0f-08	17-10	1f-18
00000000100069b0	0000000200000001	0000000400000003	0000000600000005	0000000800000007
00000000100069d0	0000000a00000009	73206f4e00006425	736b636f6c6e6970	6176612065726120
00000000100069f0	0000656c62616c69	5f7465735f717269	766973756c637865	656c646e61685f65
0000000010006a10	656b616d00000072	0068636e6172625f	625f747265736e69	6e615f68636e6172
0000000010006a30	00006b6e696c5f64	5f65766c6f736572	000068636e617262	5f6464615f717269
0000000010006a50	685f646572616873	00007265		61685f6576
0000000010006a70	5f71726900007265	6c696174		07473696c5f
0000000010006a90	2061746c65642821	0029753131203e3e	646e61685f717269	6e696168635f7265
0000000010006ab0	6c735f656572665f	20646165685f746f	0000000030203d3e	685f656c62617476
0000000010006ad0	3d2072656c646e61	61686e755f5f203d	73755f64656c646e	00007172695f7265
0000000010006af0	6e69752828282821	7629745f72747074	61685f656c626174	2d202972656c646e
0000000010006b10	70746e6975282820	71726929745f7274	72656c646e61685f	735f6e696168635f

Notice anything ... odd?

## A slide about *Gulliver's Travels*

... [a] most obstinate war ... began upon the following occasion. It is allowed on all hands, that the primitive way of breaking eggs, before we eat them, was upon the larger end; but his present majesty's grandfather, while he was a boy, going to eat an egg, and breaking it according to the ancient practice, happened to cut one of his fingers. Whereupon the emperor his father published an edict, commanding all his subjects, upon great penalties, to break the smaller end of their eggs.

Jonathan Swift, *Gulliver's Travels*

# Yet another *Gulliver's Travels* slide

There are  $10_2$  types of people who crack their eggs: one at the big end, and one at the little end.

Memory Address	Byte Value
X:	AA
X + 1:	BB
X + 2:	CC
X + 3:	DD

(a) Big-Endian

Cortex M0+	
Memory Address	Byte Value
X:	DD
X + 1:	CC
X + 2:	BB
X + 3:	AA

(b) Little-Endian

# .dis respect

program.dis (adder; with array)

10000356: b08b	sub sp, #44 ; 0x2c	MOVE DOWN IN STACK 44 BYTES OF SPACE
10000358: f003 fea2	bl 100040a0 <stdio_init_all>	
1000035c: 4669	mov r1, sp	
1000035e: 4808	ldr r0, [pc, #32] ; (10000380 <main+0x2c>)	LOAD ARRAY M <sub>0</sub>
10000360: 000b	int numbers[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}	STACK POINTER IN R3
10000362: 0002		RESS OF ARRAY IN R2
10000364: ca31		MULTIPLE, +4 AFTER
10000366: c331		STORE MULTIPLE, +4 AFTER

...

# .dis respect

program.dis (adder; for loop)

1000035a: 2400	movs r4, #0	int i = 0;
1000035c: 2c09	cmp r4, #9	
1000035e: dc05	bgt.n 1000036c <main+0x18>	i < 10;
10000360: 0021	movs r1, r4	SET CURRENT VALUE OF i IN PRINT REGISTER
10000362: 4803	ldr r0, [pc, #12] ; (10000370 <main+0x1c>)	LOAD %d
10000364: f003 fe86	bl 10004074 <__wrap_printf>	
10000368: 3401	adds r4, #1	i++;
1000036a: e7f7	b.n 1000035c <main+0x8>	}
1000036c: 2000	movs r0, #0	
1000036e: bd10	pop {r4, pc}	
10000370: 100069a0	.word 0x100069a0	

for (int i = 0; i < 10; i++) {  
    printf("%d", i);  
}



# Sizing it up

```
int main(void) {  
    stdio_init_all();  
    int sum = 0;  
    int numbers[10] = {1, 2, 3,  
                       4, 5, 6, 7,  
                       8, 9, 10};  
  
    int array_len = sizeof(numbers) / sizeof(int);  
    for(int i = 0; i < array_len; i++){  
        printf("%d (%d of %d)\n", numbers[i], i + 1, array_len);  
        sum = add(sum, numbers[i]);  
        printf("RUNNING SUM: %d\n", sum);  
    }  
    return 0;  
}
```

Python

```
numbers = [1, 2, 3  
           4, 5, 6, 7  
           8, 9, 10]
```