函数

huiw@suda.edu.cn

函数的语法

```
<return-type> functionName( <parameter-list> )
{
    <statements>
    return <expression of type return-type>;
}
```

```
int subtract(int x, int y)
{
    return x - y;
}
```

计算

$$\frac{a + a^2 + a^3}{3}$$

求三个整数的平均值

1

```c
int average (int a, int b, int c)
{
  return (a+b+c) / 3;
}
```

2

```
int
square_of(int x)
{
  return x * x;
}

int.
cube_of(int x)
{
  return x * x * x;
}
```

```
int calculate_with(int a)
{
  return average(a, square_of(a), cube_of(a));
}
```

演示代码的书写过程

求三个整数的平均值

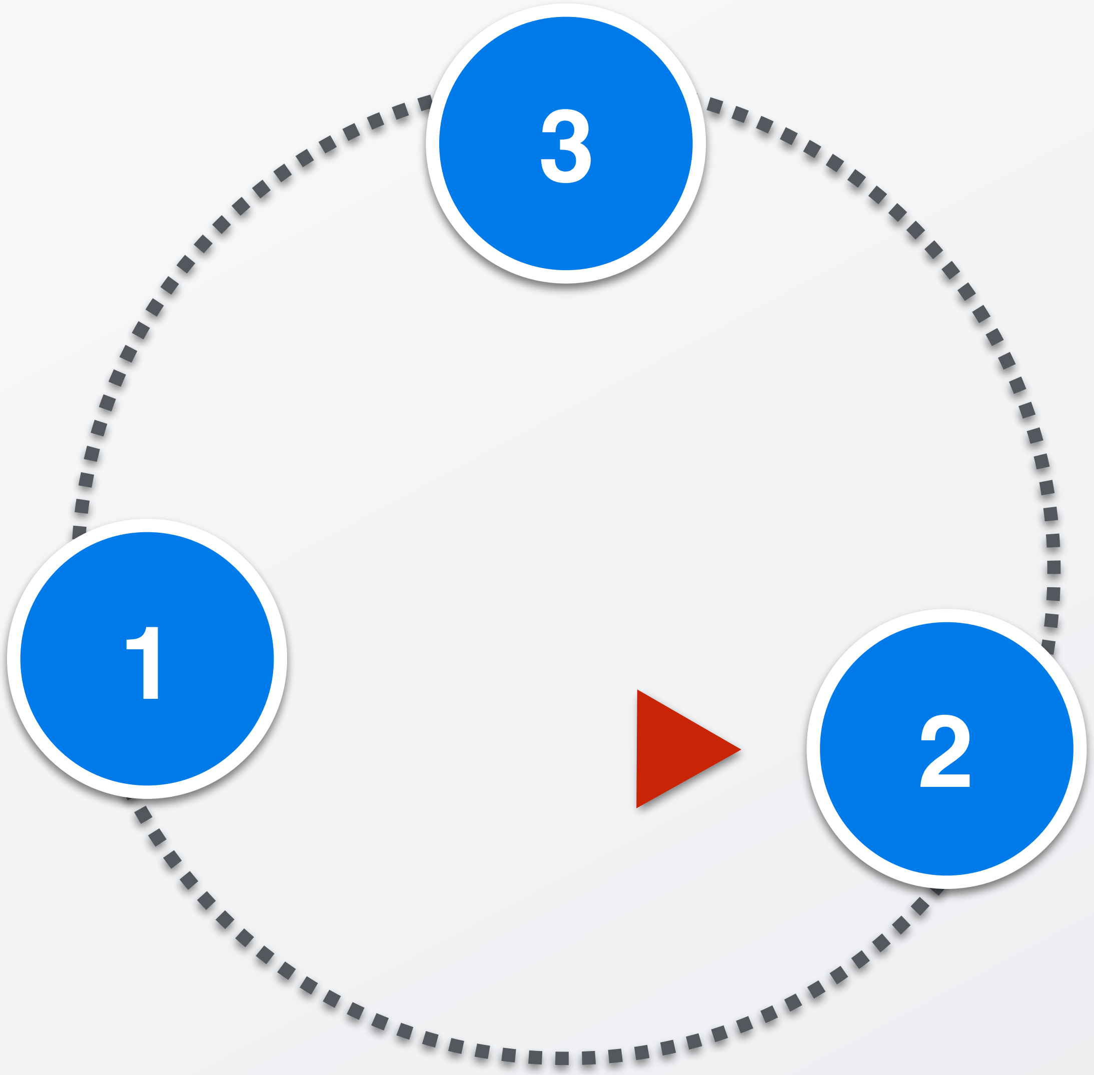
```
int average_more_numbers (int a, int b, int c)
{
  return (a+b+c) / 3;
}
```

求三个整数的平均值

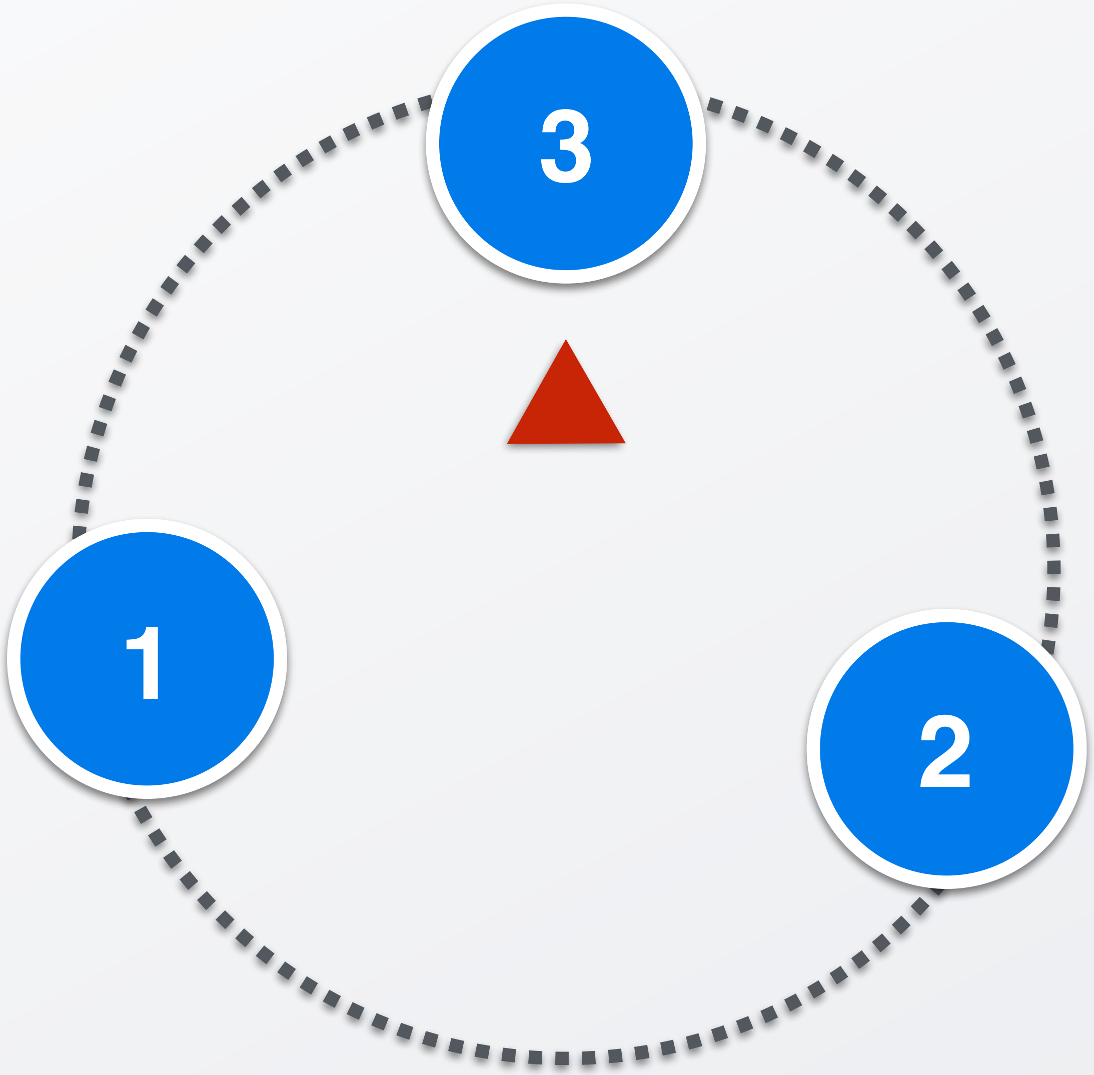
```
int average_more_numbers (int a, int b, int c)
{
  return 1 / 3 * (a+b+c);
}
```

⛔

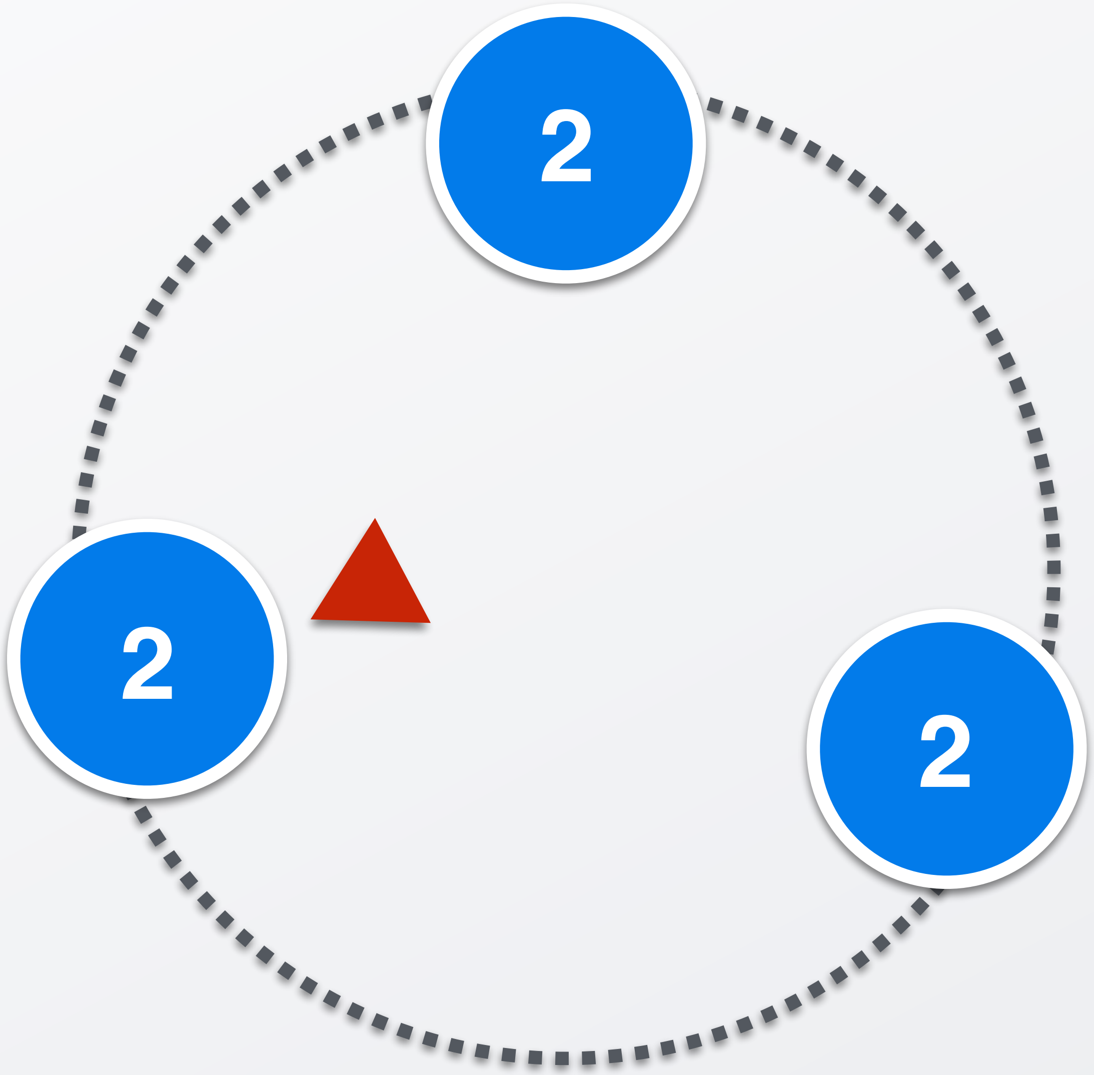
愚公移山



愚公移山



愚公移山



求三个整数的平均值

```
int average_more_numbers(int a, int b, int c)
{
  if (is_equal(a, b, c))
    return a;

  if (a > b)
    return average_more_numbers(b+1, c, a-1);
  else
    return average_more_numbers(b, c, a);
}
```

2

```
bool is_equal(int a, int b, int c)
{
  return (square_of(a-b) +.
         square_of(b-c) +.
         square_of(c-a)) <= 2;.

}
```

3

```
int square_of(int x)
{
    return x * x;
}
```

演示代码的书写过程

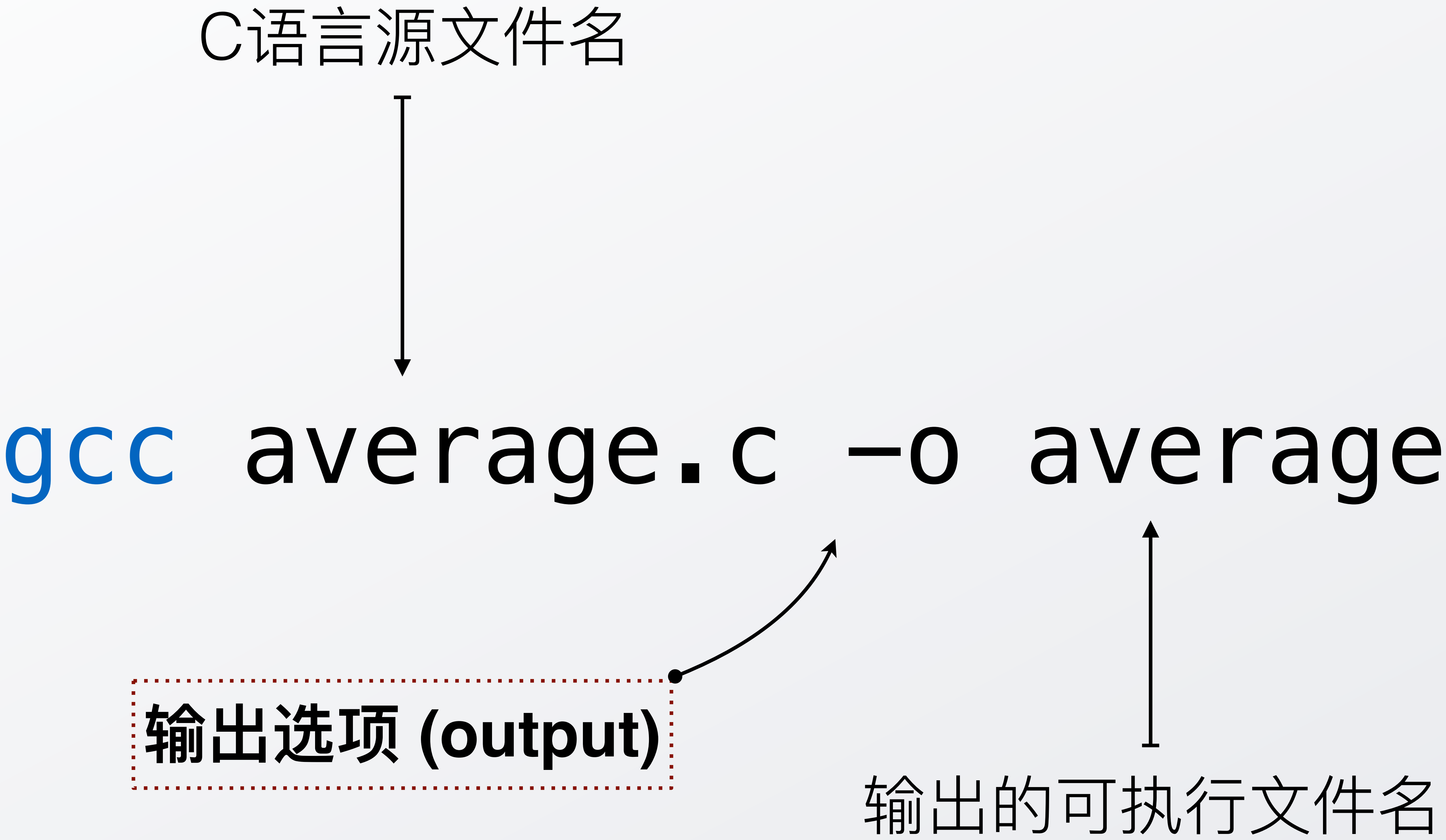
优化

消除计算结果的不确定性

```
int average_more_numbers(int a, int b, int c)
{
  if (is_equal(a, b, c))
    return minimum_of(a, b, c);

  if (a > b)
    return average_more_numbers(b+1, c, a-1);
  else
    return average_more_numbers(b, c, a);
}
```

```c
int min(int a, int b)
{
  if (a < b)
    return a;

  return b;
}

int minimum(int a, int b, int c)
{
  return min(a, min(b, c));
}
```

使用编译器



C语言源文件名

```
gcc average.c -o average
```

输出选项 (output)

输出的可执行文件名

```
gcc -g average.c -o average
```

调试选项 (debug)

函数声明

```
<return-type> functionName( <parameter-list> );
```

注意这里的分号

```
bool is_equal(int a, int b, int c);
```

注：函数声明也称为函数原型。

函数定义

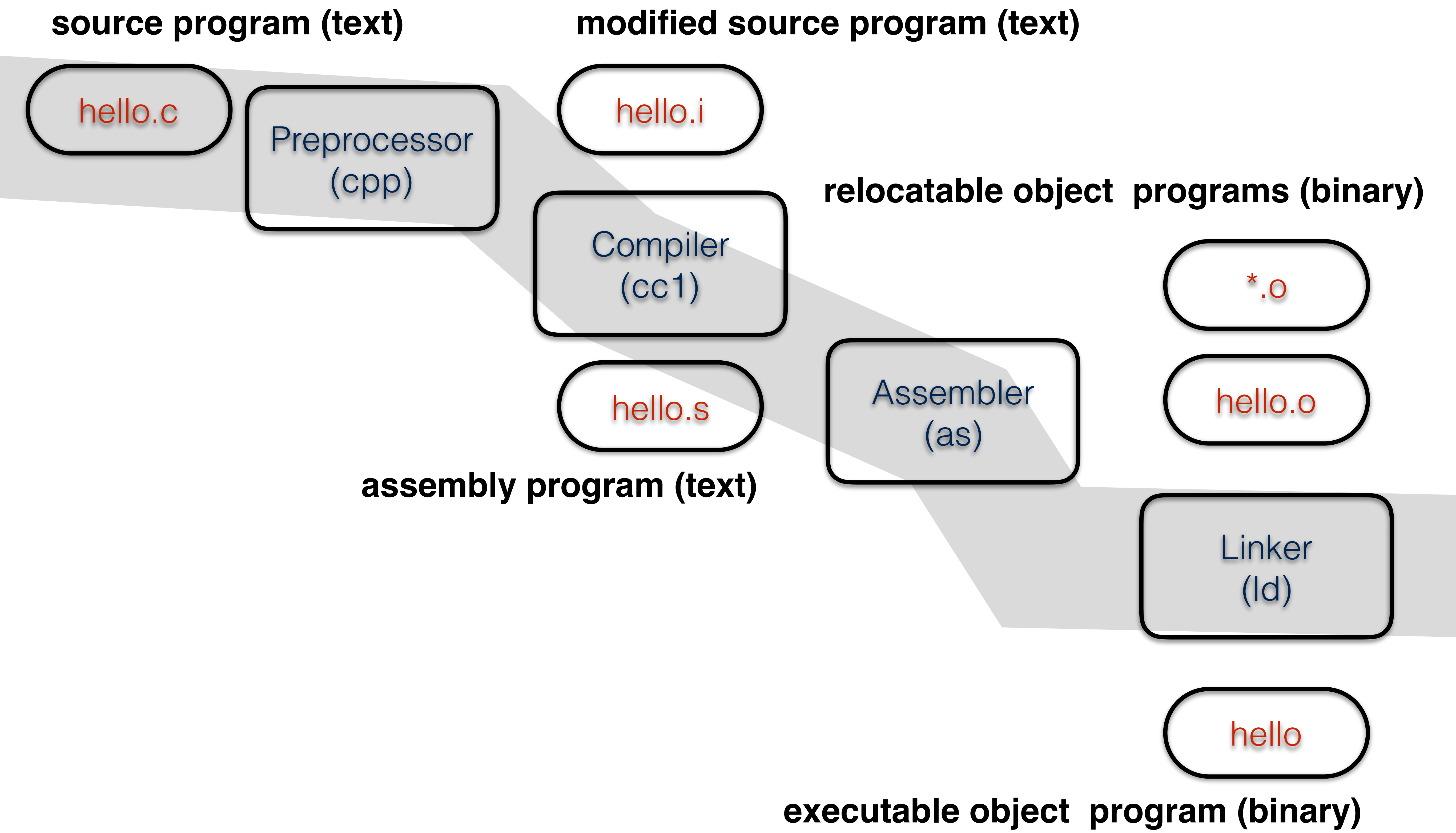
```
bool is_equal(int a, int b, int c)
{
  return (square(a-b) +.
         square(b-c) +.
         square(c-a)) <= 2;.

}
```

程序执行入口

```
int main (int argc, char* argv[]);
int main ();
```

C语言规定程序启动后会调用命名为**main**的函数。

预处理、编译与链接



source program (text)

hello.c

Preprocessor
(cpp)

modified source program (text)

hello.i

Compiler
(cc1)

hello.s

assembly program (text)

Assembler
(as)

relocatable object programs (binary)

*.o

hello.o

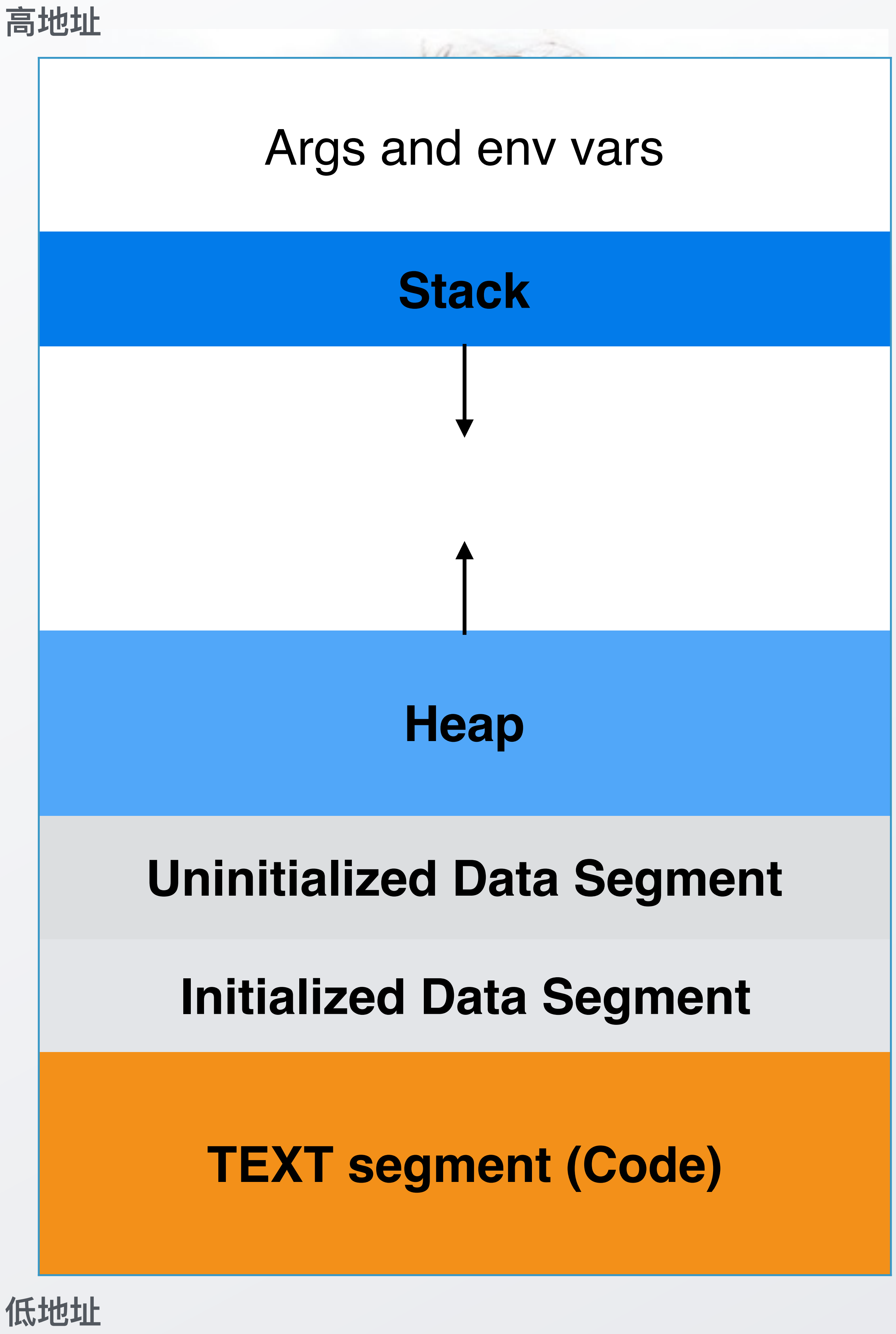
Linker
(ld)

hello

executable object program (binary)

程序执行

存储器布局


高地址
Args and env vars
Stack
Heap
Uninitialized Data Segment
Initialized Data Segment
TEXT segment (Code)
低地址

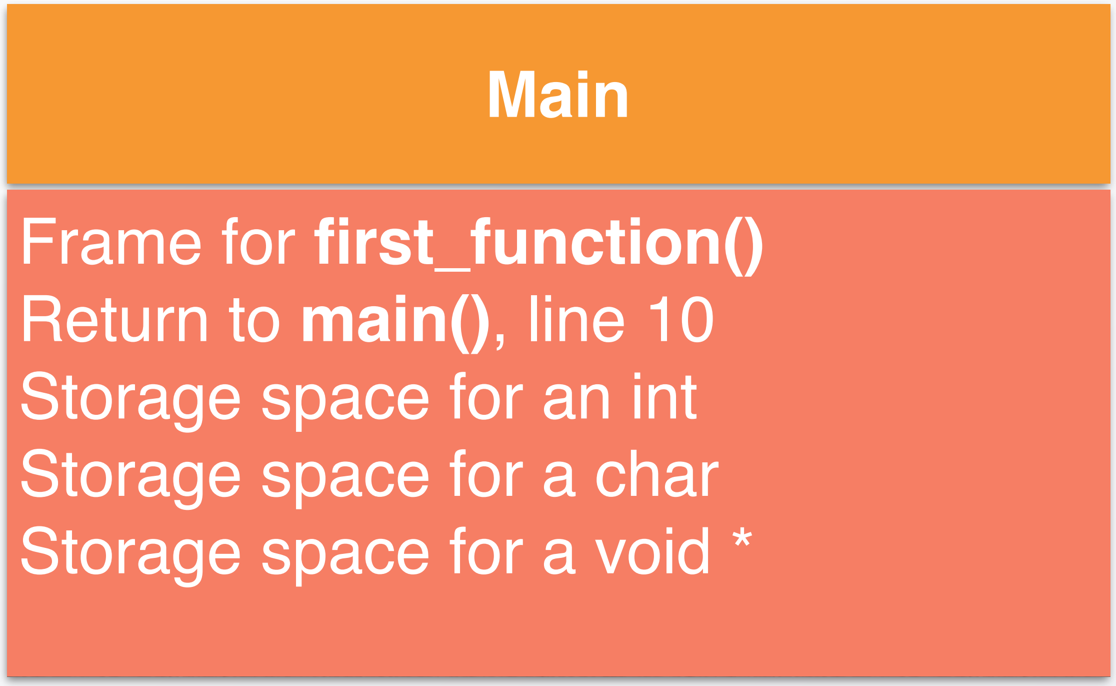
Main

```c
#include <stdio.h>

void first_function(void);
int second_function(int);

int main(void)
{
    printf("hello world\n");
    first_function();
    printf("goodbye goodbye\n");

    return 0;
}


void first_function(void)
{
    int imidate = 3;
    char broiled = 'c';
    void *where_prohibited = NULL;

    second_function(imidate);
    imidate = 10 + (int)broiled;                // junk codes
    where_prohibited = (void *)&imidate;
}


int second_function(int a)
{
    int b = a;
    return b;
}
```

Main

```c
#include <stdio.h>

void first_function(void);
int second_function(int);

int main(void)
{
    printf("hello world\n");
    first_function();
    printf("goodbye goodbye\n");

    return 0;
}


void first_function(void)
{
    int imidate = 3;
    char broiled = 'c';
    void *where_prohibited = NULL;

    second_function(imidate);
    imidate = 10 + (int)broiled;              // junk codes
    where_prohibited = (void *)&imidate;
}


int second_function(int a)
{
    int b = a;
    return b;
}
```

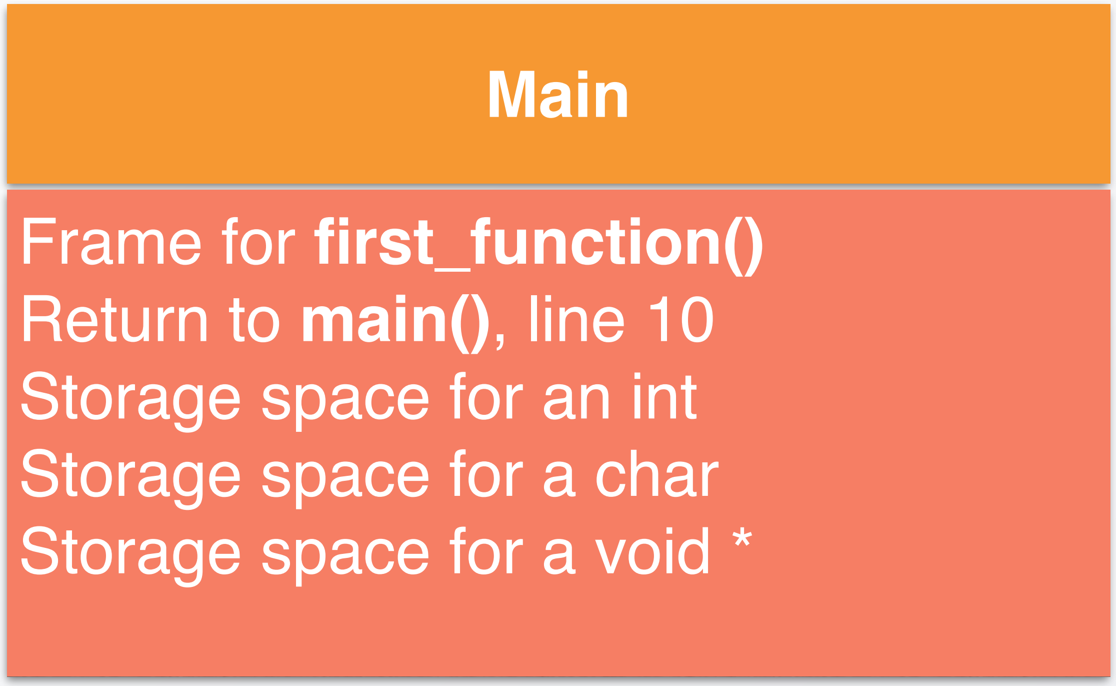
```c
 1 #include <stdio.h>
 2
 3 void first_function(void);
 4 int second_function(int);
 5
 6 int main(void)
 7 {
 8     printf("hello world\n");
 9     first_function();
10     printf("goodbye goodbye\n");
11
12     return 0;
13 }
14
15
16 void first_function(void)
17 {
18     int imidate = 3;
19     char broiled = 'c';
20     void *where_prohibited = NULL;
21
22     second_function(imidate);
23     imidate = 10 + (int)broiled;                  // junk codes
24     where_prohibited = (void *)&imidate;
25 }
26
27
28 int second_function(int a)
29 {
30     int b = a;
31     return b;
32 }
```



Main

Frame for **first_function()**
Return to **main()**, line 10
Storage space for an int
Storage space for a char
Storage space for a void *

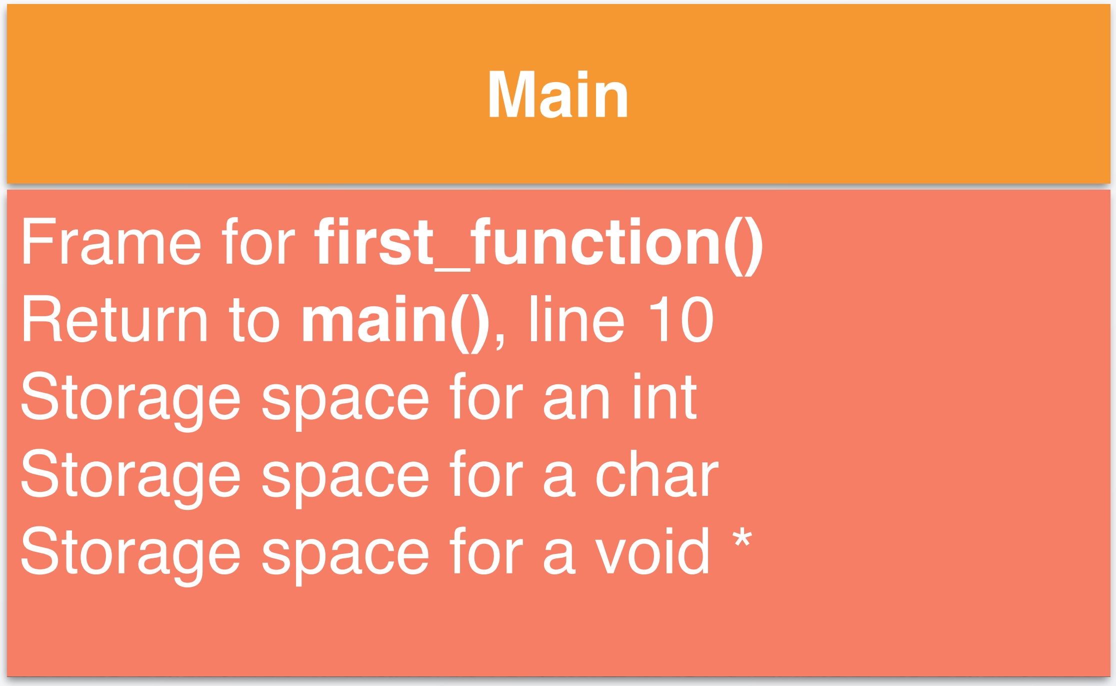
```c
 1 #include <stdio.h>
 2
 3 void first_function(void);
 4 int second_function(int);
 5
 6 int main(void)
 7 {
 8     printf("hello world\n");
 9     first_function();
10     printf("goodbye goodbye\n");
11
12     return 0;
13 }
14
15
16 void first_function(void)
17 {
18     int imidate = 3;
19     char broiled = 'c';
20     void *where_prohibited = NULL;
21
22     second_function(imidate);
23     imidate = 10 + (int)broiled;                 // junk codes
24     where_prohibited = (void *)&imidate;
25 }
26
27
28 int second_function(int a)
29 {
30     int b = a;
31     return b;
32 }
```



Main

Frame for **first_function()**
Return to **main()**, line 10
Storage space for an int
Storage space for a char
Storage space for a void *

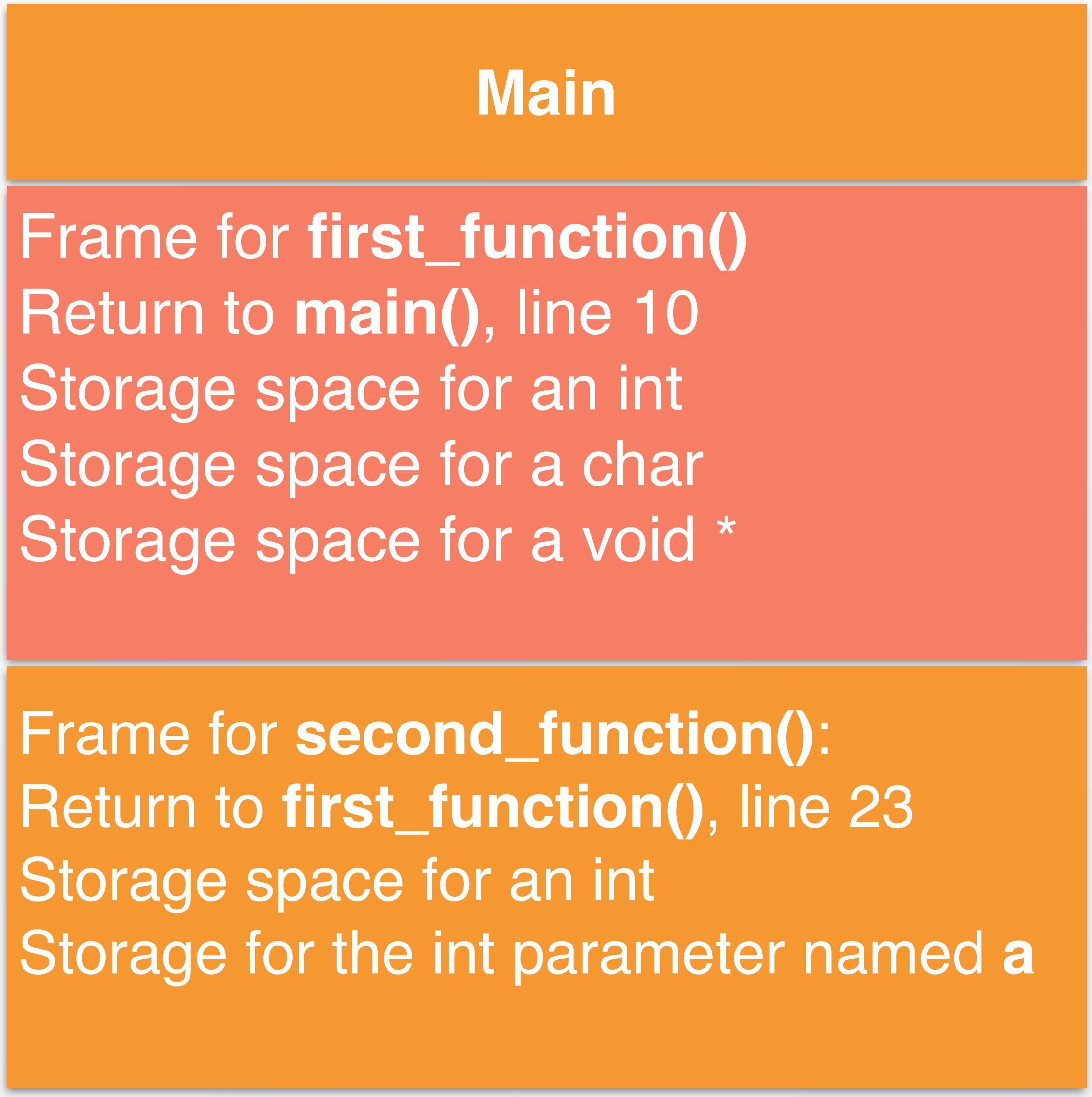
```
 1 #include <stdio.h>
 2
 3 void first_function(void);
 4 int second_function(int);
 5
 6 int main(void)
 7 {
 8    printf("hello world\n");
 9    first_function();
10    printf("goodbye goodbye\n");
11
12    return 0;
13 }
14
15
16 void first_function(void)
17 {
18    int imidate = 3;
19    char broiled = 'c';
20    void *where_prohibited = NULL;
21
22    second_function(imidate);
23    imidate = 10 + (int)broiled;                // junk codes
24    where_prohibited = (void *)&imidate;
25 }
26
27
28 int second_function(int a)
29 {
30    int b = a;
31    return b;
32 }
```



Main

Frame for **first_function()**
Return to **main()**, line 10
Storage space for an int
Storage space for a char
Storage space for a void *

```c
 1 #include <stdio.h>
 2
 3 void first_function(void);
 4 int second_function(int);
 5
 6 int main(void)
 7 {
 8     printf("hello world\n");
 9     first_function();
10     printf("goodbye goodbye\n");
11
12     return 0;
13 }
14
15
16 void first_function(void)
17 {
18     int imidate = 3;
19     char broiled = 'c';
20     void *where_prohibited = NULL;
21
22     second_function(imidate);
23     imidate = 10 + (int)broiled;
24     where_prohibited = (void *)&imidate;
25 }
26
27
28 int second_function(int a)
29 {
30     int b = a;
31     return b;
32 }
```



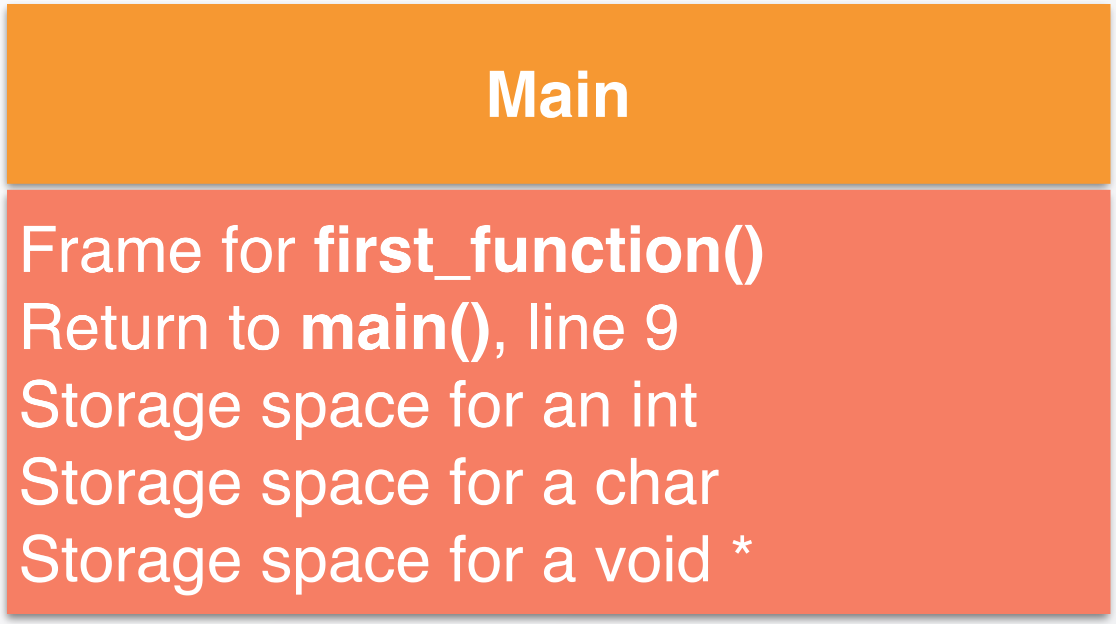
Main

Frame for **first_function()**
Return to **main()**, line 10
Storage space for an int
Storage space for a char
Storage space for a void *

Frame for **second_function()**:
Return to **first_function()**, line 23
Storage space for an int
Storage for the int parameter named **a**

```
// junk codes
```

```
 1 #include <stdio.h>
 2
 3 void first_function(void);
 4 int second_function(int);
 5
 6 int main(void)
 7 {
 8     printf("hello world\n");
 9     first_function();
10     printf("goodbye goodbye\n");
11
12     return 0;
13 }
14
15
16 void first_function(void)
17 {
18     int imidate = 3;
19     char broiled = 'c';
20     void *where_prohibited = NULL;
21
22     second_function(imidate);
23     imidate = 10 + (int)broiled;                 // junk codes
24     where_prohibited = (void *)&imidate;
25 }
26
27
28 int second_function(int a)
29 {
30     int b = a;
31     return b;
32 }
```



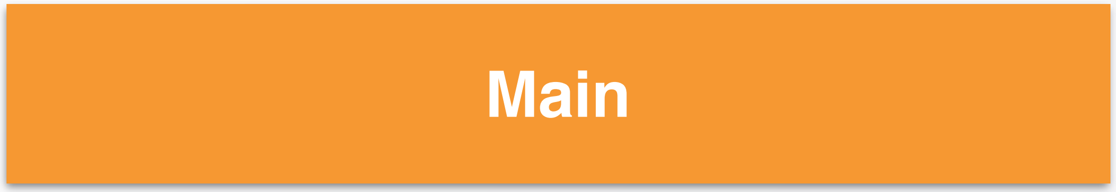
Main

Frame for **first_function()**
Return to **main()**, line 9
Storage space for an int
Storage space for a char
Storage space for a void *

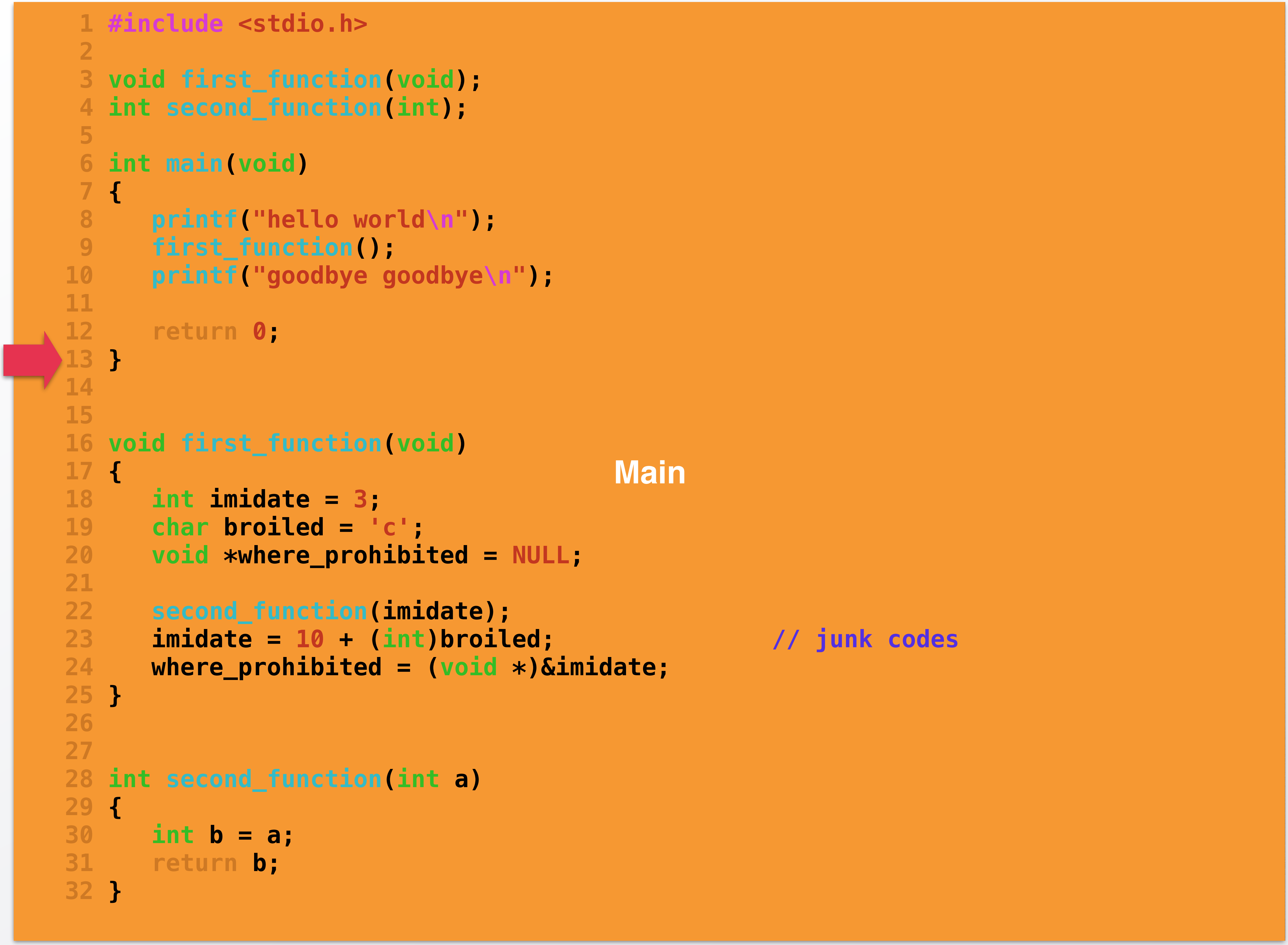
Main

```c
#include <stdio.h>

void first_function(void);
int second_function(int);

int main(void)
{
    printf("hello world\n");
    first_function();
    printf("goodbye goodbye\n");

    return 0;
}


void first_function(void)
{
    int imidate = 3;
    char broiled = 'c';
    void *where_prohibited = NULL;

    second_function(imidate);
    imidate = 10 + (int)broiled;              // junk codes
    where_prohibited = (void *)&imidate;
}


int second_function(int a)
{
    int b = a;
    return b;
}
```

```c
#include <stdio.h>

void first_function(void);
int second_function(int);

int main(void)
{
    printf("hello world\n");
    first_function();
    printf("goodbye goodbye\n");

    return 0;
}


void first_function(void)
{
    int imidate = 3;
    char broiled = 'c';
    void *where_prohibited = NULL;

    second_function(imidate);
    imidate = 10 + (int)broiled;                     // junk codes
    where_prohibited = (void *)&imidate;
}


int second_function(int a)
{
    int b = a;
    return b;
}
```



Main



```c
#include <stdio.h>

void first_function(void);
int second_function(int);

int main(void)
{
    printf("hello world\n");
    first_function();
    printf("goodbye goodbye\n");

    return 0;
}


void first_function(void)
{
    int imidate = 3;
    char broiled = 'c';
    void *where_prohibited = NULL;

    second_function(imidate);
    imidate = 10 + (int)broiled;                // junk codes
    where_prohibited = (void *)&imidate;
}


int second_function(int a)
{
    int b = a;
    return b;
}
```

Main

END

课堂练习

$$f(x) = 3x^5 + 2x^4 - 5x^3 - x^2 + 7x - 6, \quad x \in \mathbb{Z}$$

请编程实现函数$f(x)$。

课堂练习

$$\begin{aligned} f(x) =& 3x^5 + 2x^4 - 5x^3 - x^2 + 7x - 6 \\ =& (3x^4 + 2x^3 - 5x^2 - x + 7)x - 6 \\ =& ((3x^3 + 2x^2 - 5x - 1)x + 7)x - 6 \\ =& (((3x^2 + 2x - 5)x - 1)x + 7)x - 6 \\ =& ((((3x + 2)x - 5)x - 1)x + 7)x - 6, \\ & x \in \mathbb{Z} \end{aligned}$$

请编程实现函数$f(x)$。

WENZHENG COLLEGE OF SOOCHOW UNIVERSITY

2017.3.29

Soochow University