Computer Science & IT

C programming

Functon & Storage Class

Lecture No. 01

By- Abhishek Sir

Recap of Previous Lecture

Topic	do while
Topic	break
Topic	Continue
Topic	function begin.
Topic	

Slide

Topics to be Covered

Topic function

Topic Call by value

Topic Storage class (Activation Records)

Topic

Topic

Slide

Engineering

Aptitude + Maths + Discrete Maths

33%

function

Recursion

Function

```
# include<stdio.h>
int fun(int, int);
int main() {
        int a=10, b=20;
        int k;
        k = fun(a,b)
        printf("%d", k)
        return 0;
}
```

function declaration

function call

30

function definition

i=10, j=20

i=a, j=b

```
int fun(int i, int j) {
    return i+j;
}
```

10+20

* a, b arguments, actual parameter

Call by value

* i, j parameter

formal parameter

Slide

Function

- * if function is called control transfer from main function to called function.
- * Save the information of main function then control is transferred.
- * Actual parameter value will be copied to formal parameter; (Call by value)
- * function will execute. after termination control will transfer to main function

Slide

Function

$\boxed{Xy}$

xy

Control → Sequential flow of execution

```
# include<stdio.h>
void foo(int, int);
int main(){
    int a=10, b=20;  ✓
    foo(a,b);            10,20
    printf("%d %d", a,b);
    return 0;
}
```

a=10, b=20

```
void foo (int a, int b){
swap  int temp;
      temp=a;        temp=10
      a=b;           a=20
      b=temp;        b=10
      printf("%d %d", a,b);
                     20 10
}
```

Slide

Function

* main local variable a,b is different from Local variable a,b of fou()

* Local value will be swapped.

*

Slide

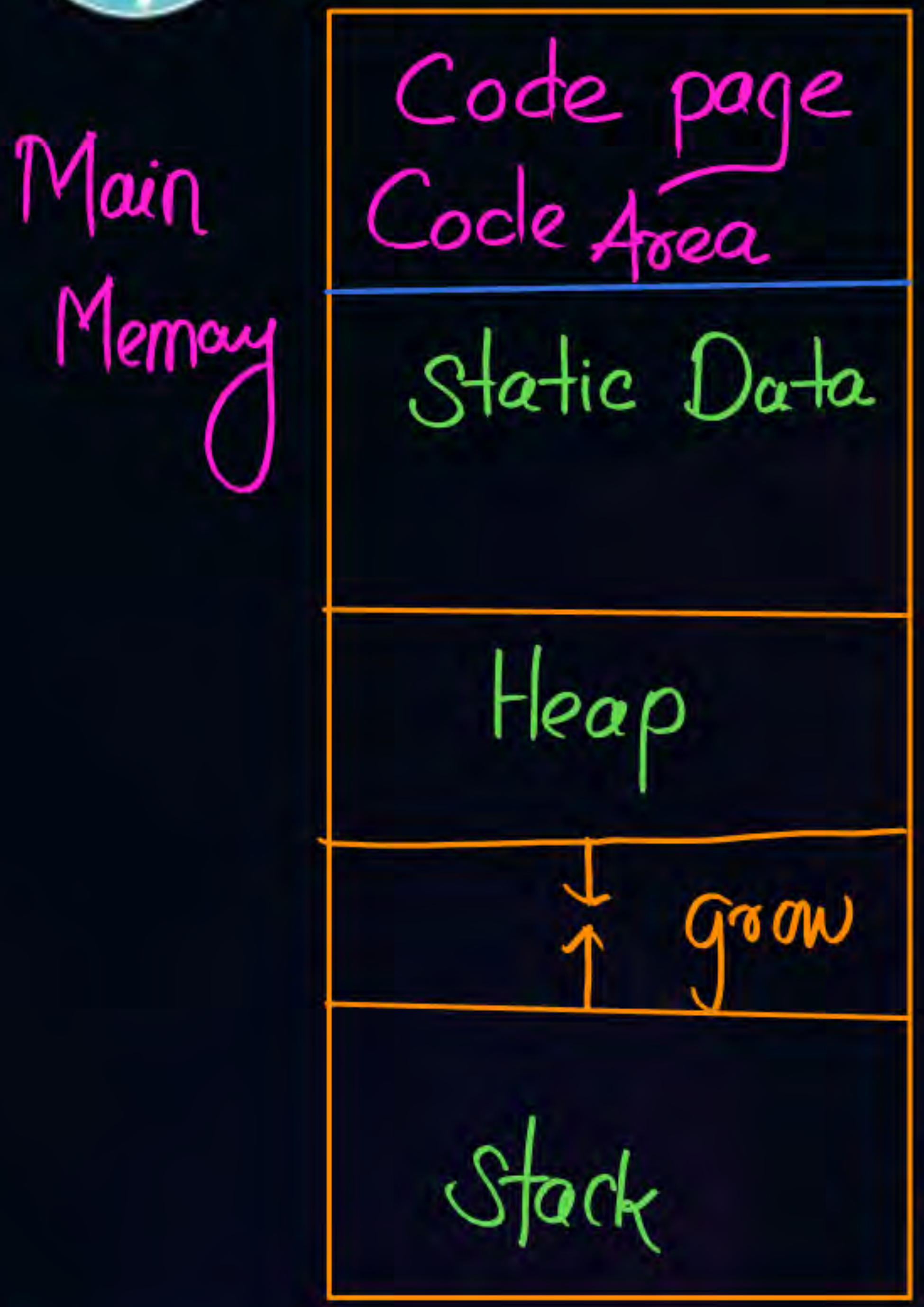
Function

Activation Record : When a function (procedure) is in Execution, the information Regarding execution of function stored as Record (Structure) calleds Activation Record

- * Local variable ✓
- * Machine status (PSW) program status word
  flag, carry, overflow, zero
- * control links . . . . ect.

Slide

Function

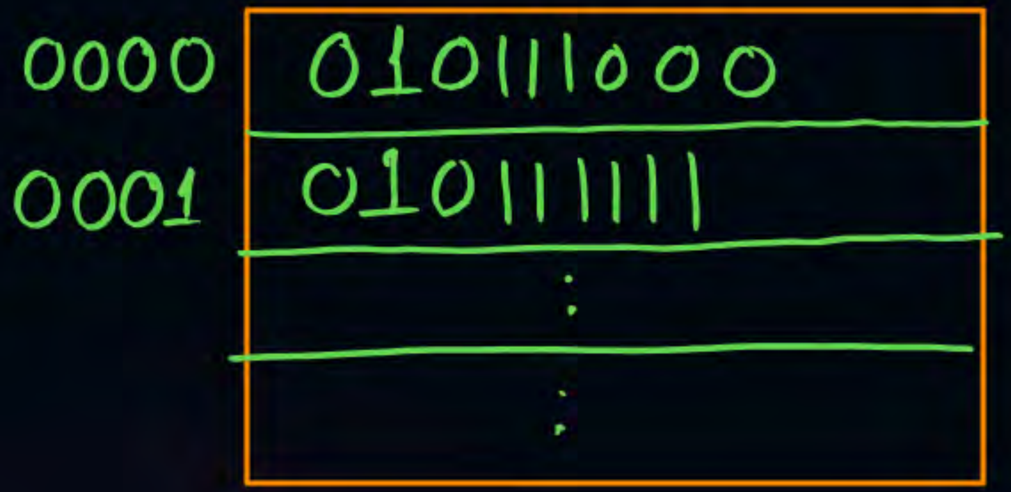


Memory Layout : When a program is in execution different Area of memory Required.

Code : Exe-file (string of 0's and 1's) Loaded by loader for execution in Code page or code Area.

↙ local variable stack.

Function



0000 010111000

0001 010111111

:

:

Code Area

Slide

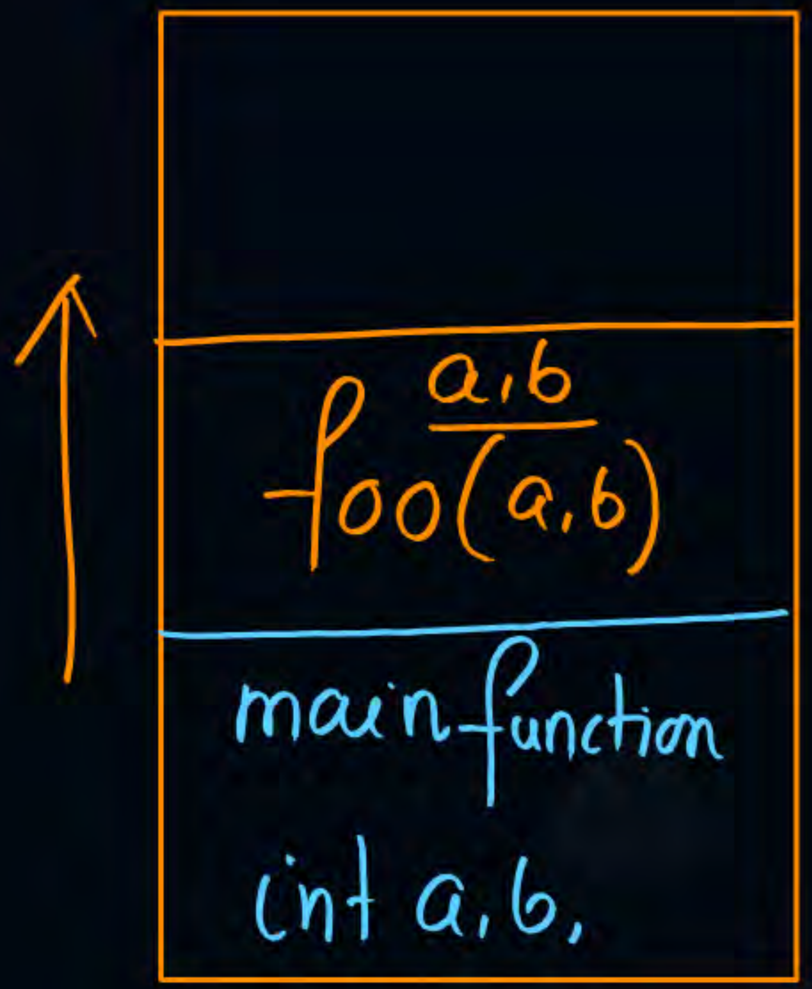
Function

Local variable ← Activation Record

if a function is in Execution

the Activation Record is created

push in Stack Area of Memory



Last in first out one ended data structure

Slide

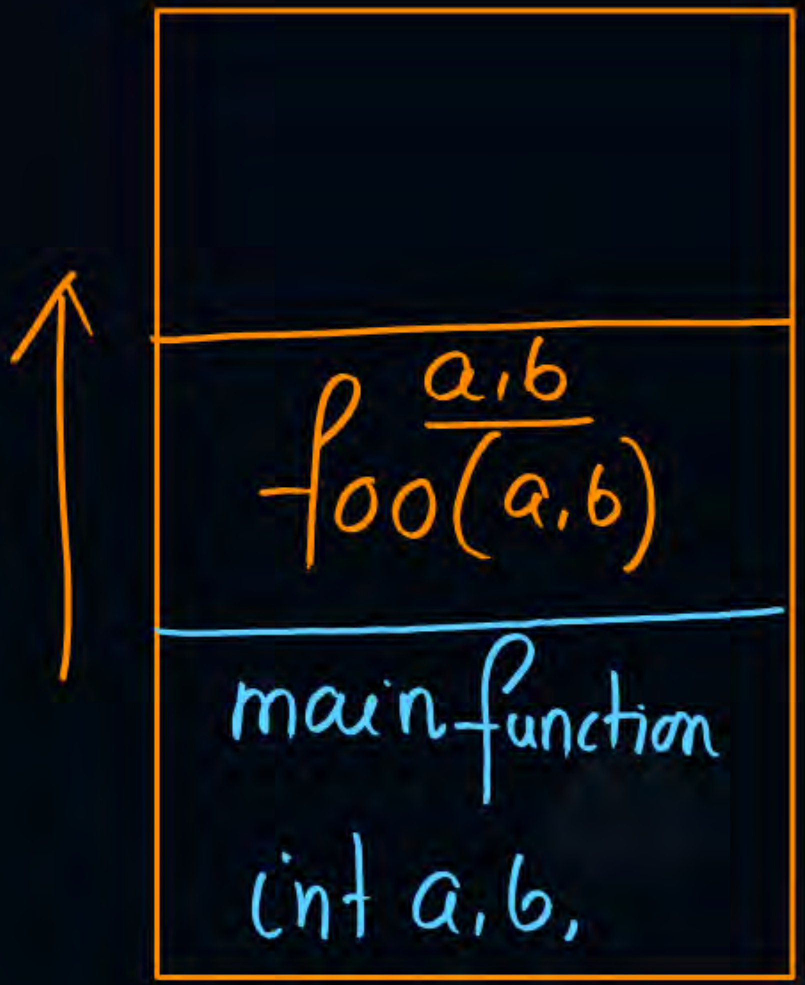
Function

As foo function Terminates

* Activation Record will be deleted (pop) from stack.

* If Activation Record is deleted from stack what will happen to local variable.

Delete / Destroy / Deallocated



(Stack Runtime)

Last in first out one ended data structure

Slide

Function

```
int main() {
    ⋮
    f1()
    f3()
    ⋮
}
```

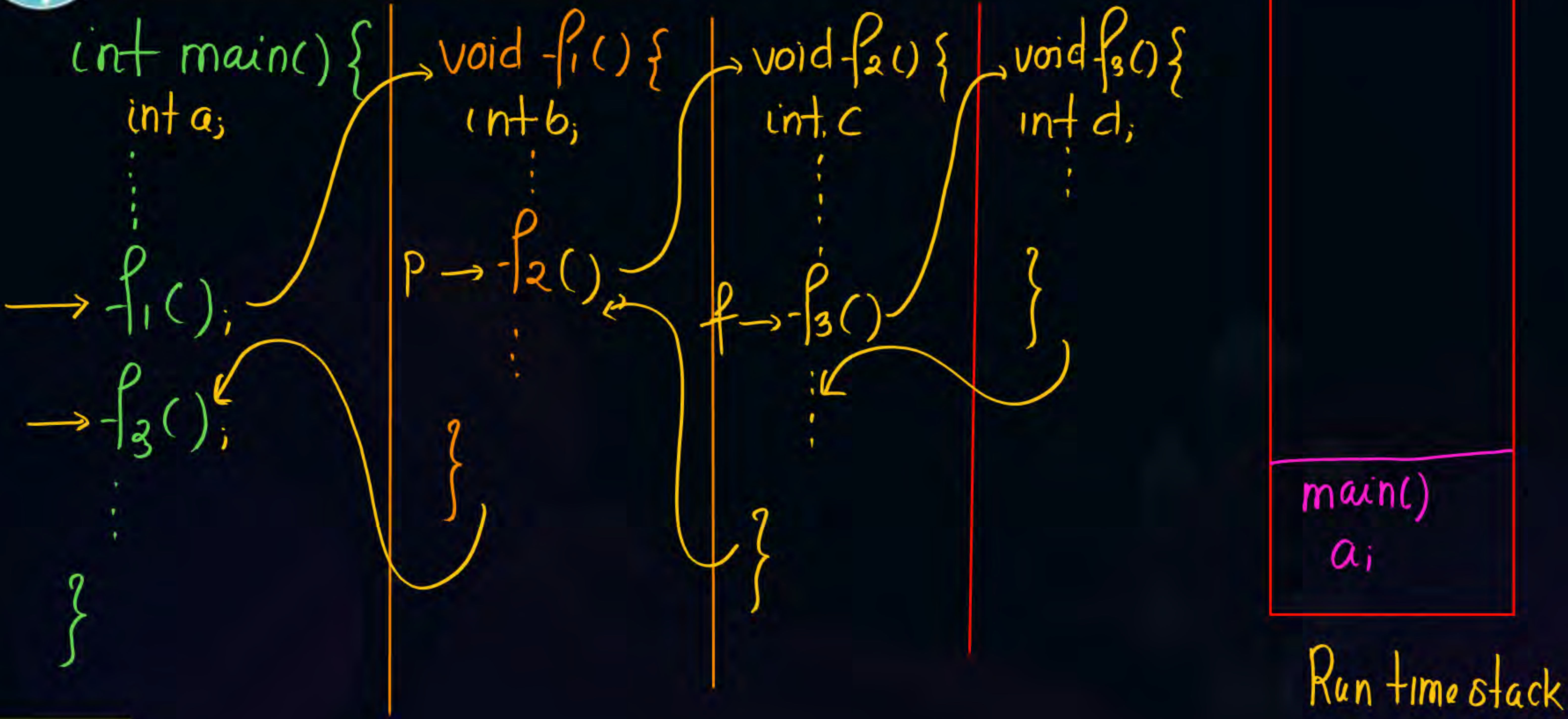
```
void f1() {
    ⋮
    f2()
    ⋮
}
```

```
void f2() {
    ⋮
    f3()
    ⋮
}
```

```
void f3() {
    ⋮
}
```

Slide

Function



```
int main() {
    int a;
      :
  →  f1();
  →  f3();
      :
}
```

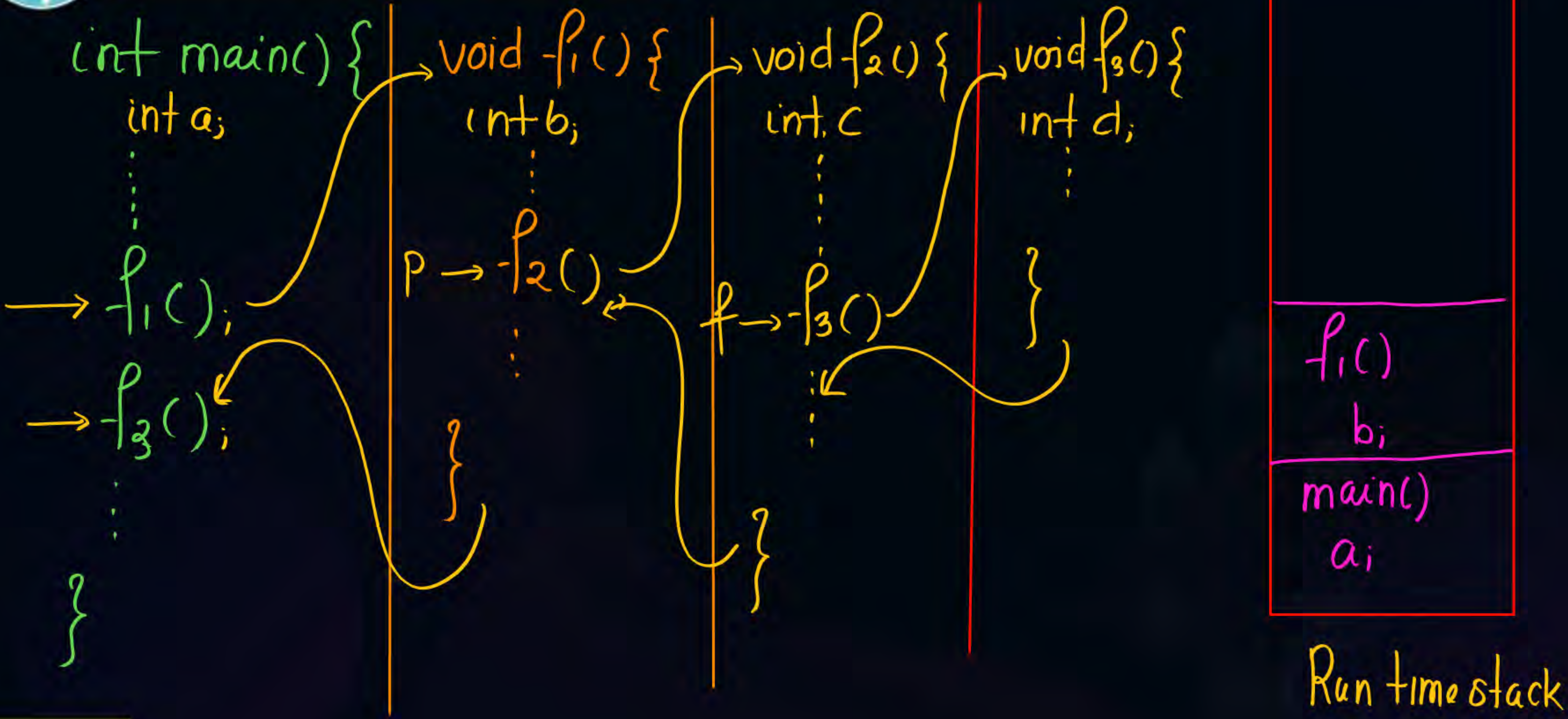
```
void f1() {
    int b;
      :
  p →  f2()
      :
}
```

```
void f2() {
    int c
      :
  f →  f3()
      :
}
```

```
void f3() {
    int d;
      :
}
```

Slide

Function



```
int main() {
    int a;
    :
 → f1();
 → f3();
    :
}
```

```
void f1() {
    int b;
    :
 p → f2()
    :
}
```

```
void f2() {
    int c
    :
 f → f3()
    :
}
```

```
void f3() {
    int d;
    :
}
```

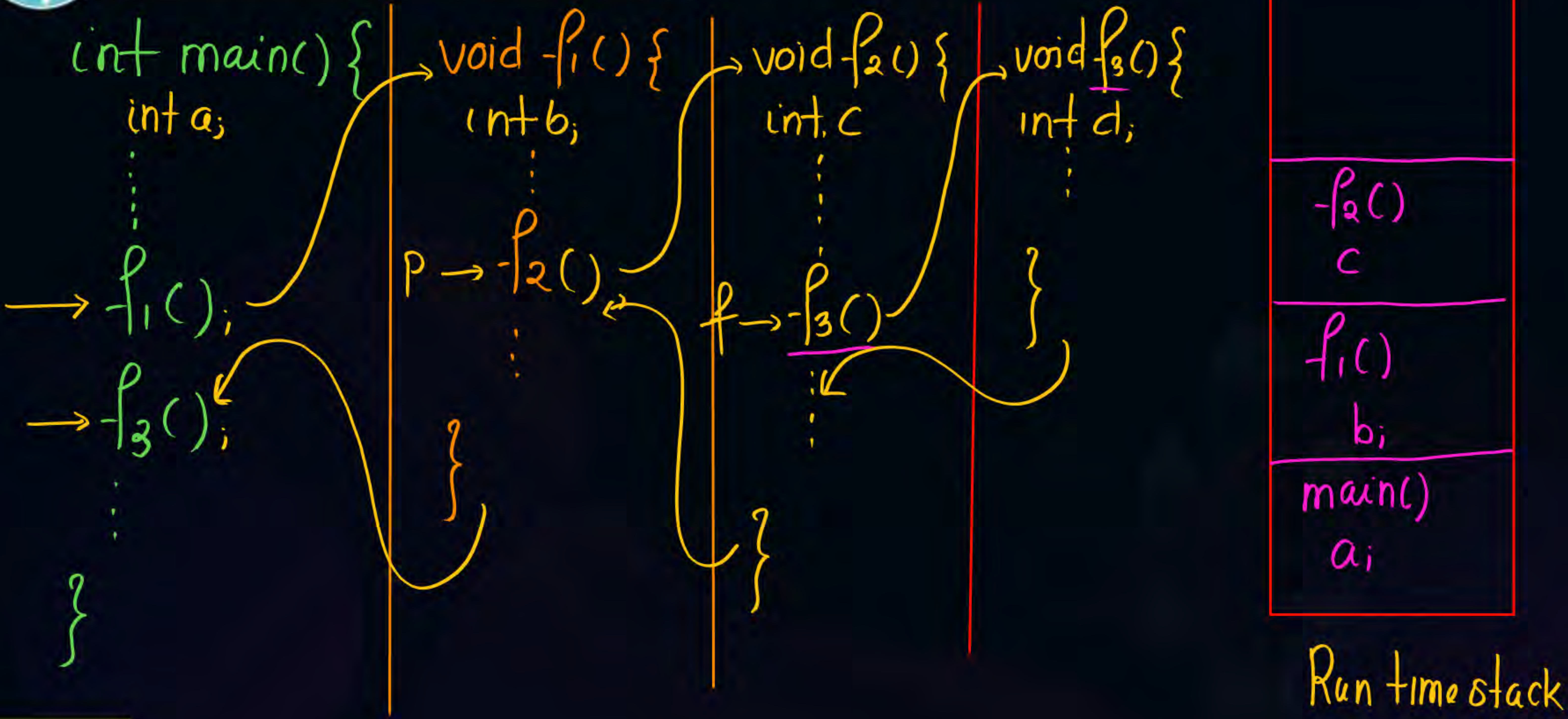
f1()
b;

main()
a;

Run time stack

Slide

Function



```
int main() {
    int a;
    ⋮
→   f1();
→   f3();
    ⋮
}
```

```
void f1() {
    int b;
    ⋮
p → f2()
    ⋮
}
```

```
void f2() {
    int c
    ⋮
f → f3()
    ⋮
}
```

```
void f3() {
    int d;
    ⋮
}
```

f2() c

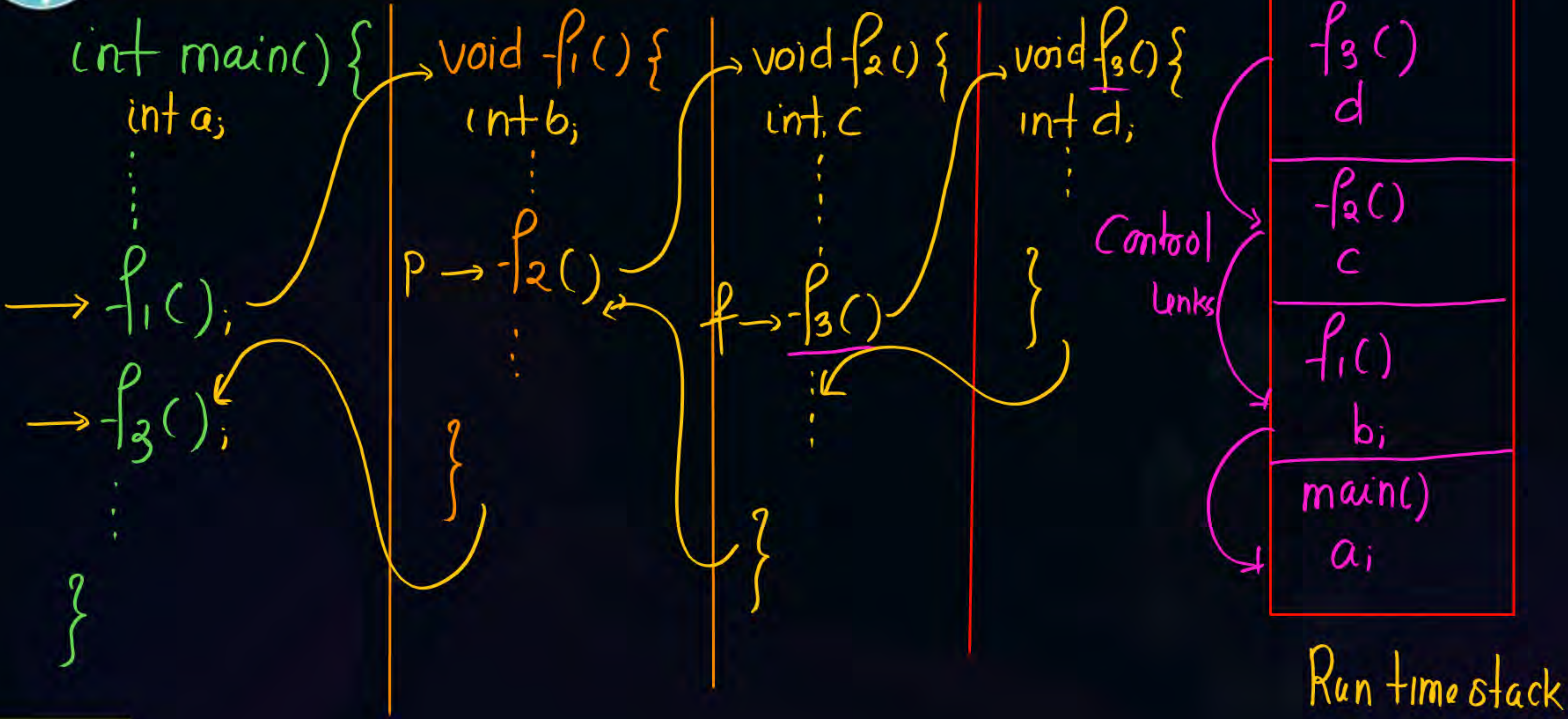
f1()
b;

main()
a;

Run time stack

Slide

Function



```
int main() {
    int a;
    ⋮
→ f1();
→ f3();
    ⋮
}
```

```
void f1() {
    int b;
    ⋮
p → f2()
    ⋮
}
```

```
void f2() {
    int. c
    ⋮
f → f3()
    ⋮
}
```

```
void f3() {
    int d;
    ⋮
}
```

f3()
d

f2()
c

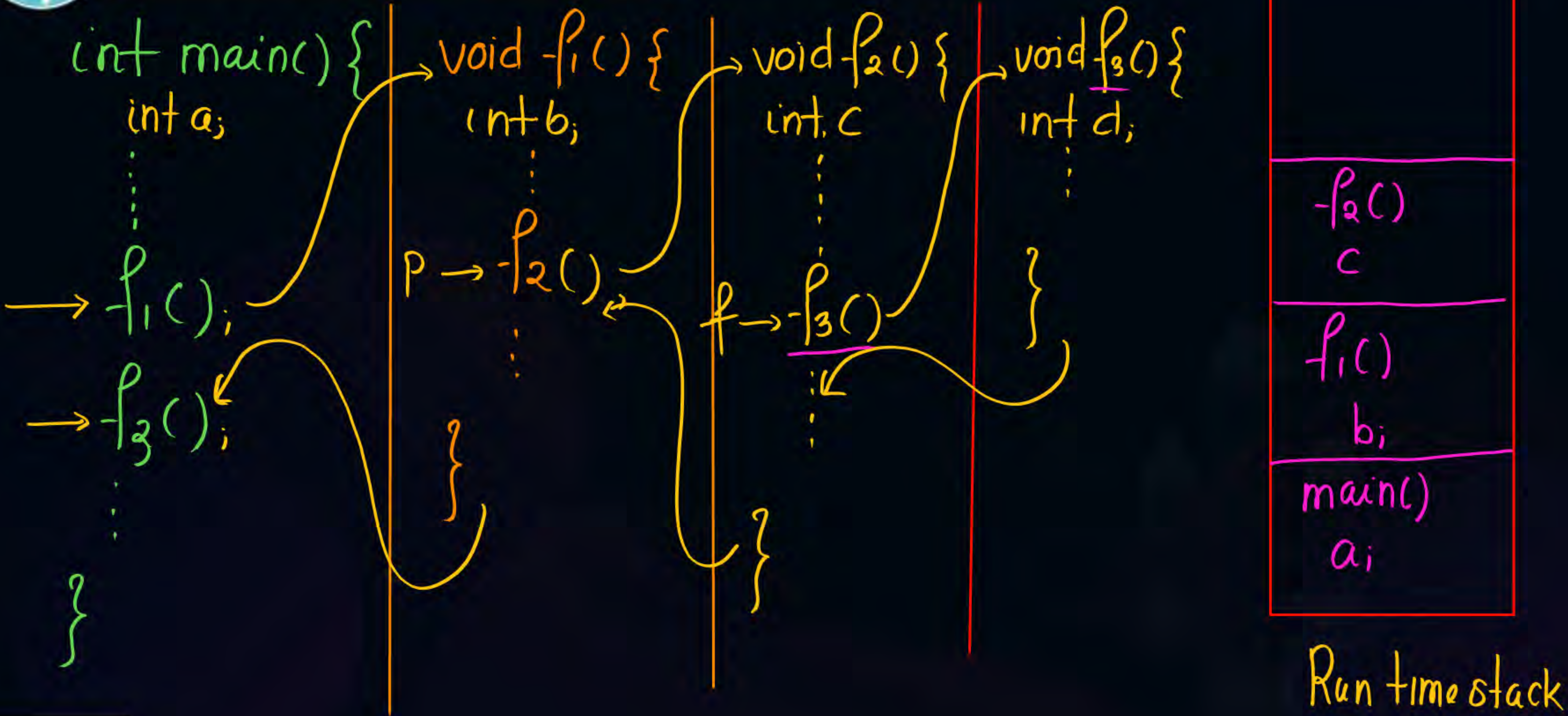
f1()
b;

main()
a;

Run time stack

Slide

Function



```
int main() {
    int a;
    ⋮
→   f1();
→   f3();
    ⋮
}
```

```
void f1() {
    int b;
    ⋮
p → f2()
    ⋮
}
```

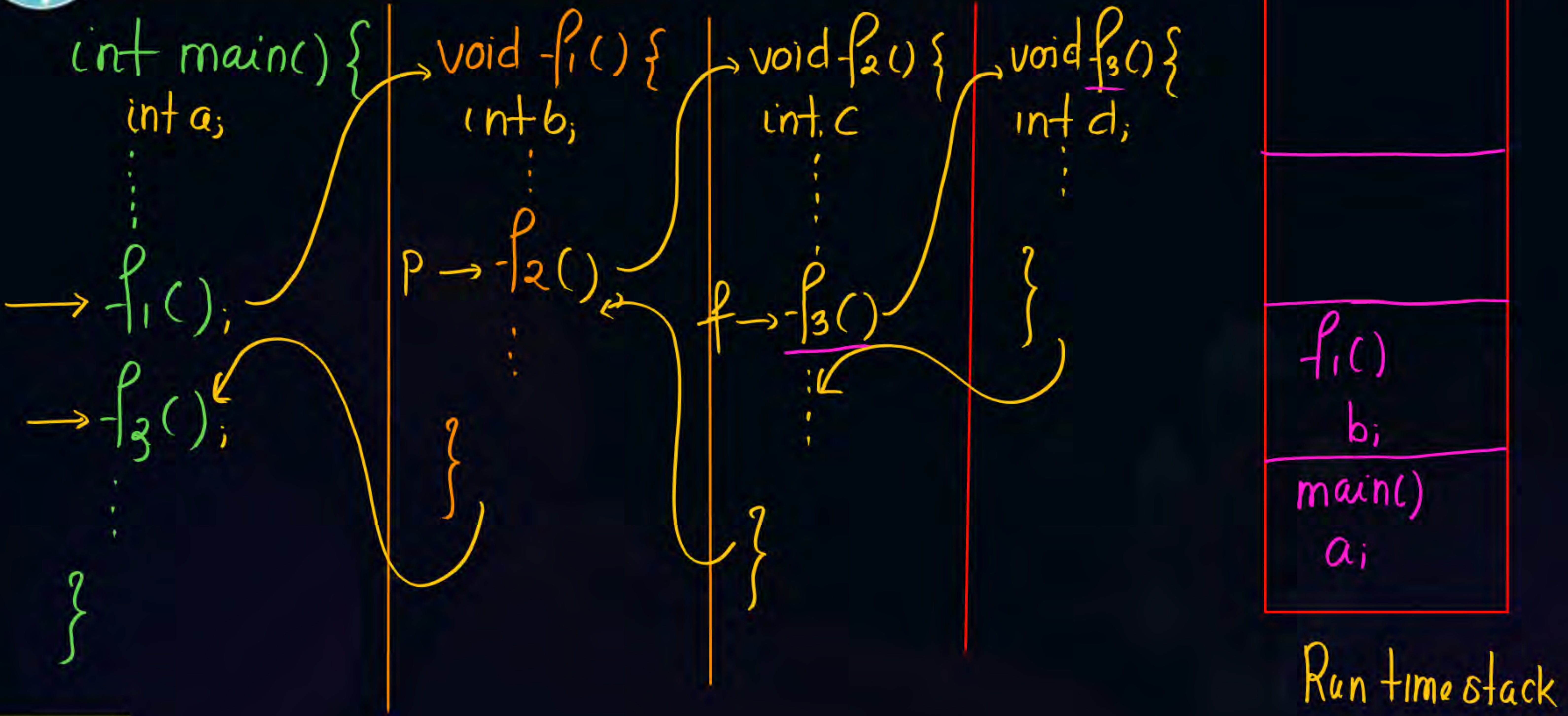
```
void f2() {
    int c
    ⋮
f → f3()
    ⋮
}
```

```
void f3() {
    int d;
    ⋮
}
```

Run time stack

Slide

Function



```
int main() {
    int a;
    ⋮
→ f1();
→ f3();
    ⋮
}
```

```
void f1() {
    int b;
    ⋮
p → f2()
    ⋮
}
```

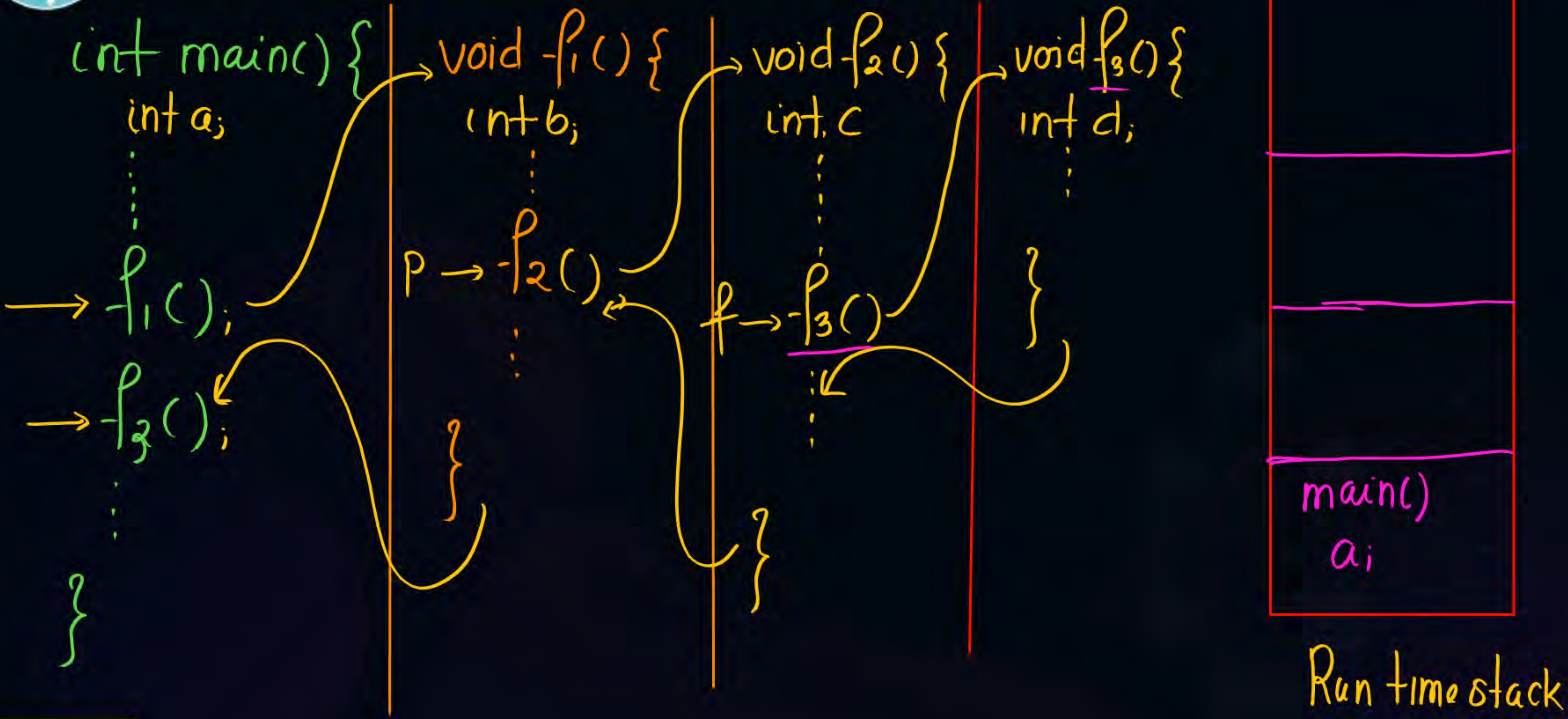
```
void f2() {
    int. c
    ⋮
f → f3()
    ⋮
}
```

```
void f3() {
    int d;
    ⋮
}
```

Run time stack

Slide

Function



```
int main() {
    int a;
    :
  → f1();
  → f2();
    :
}
```

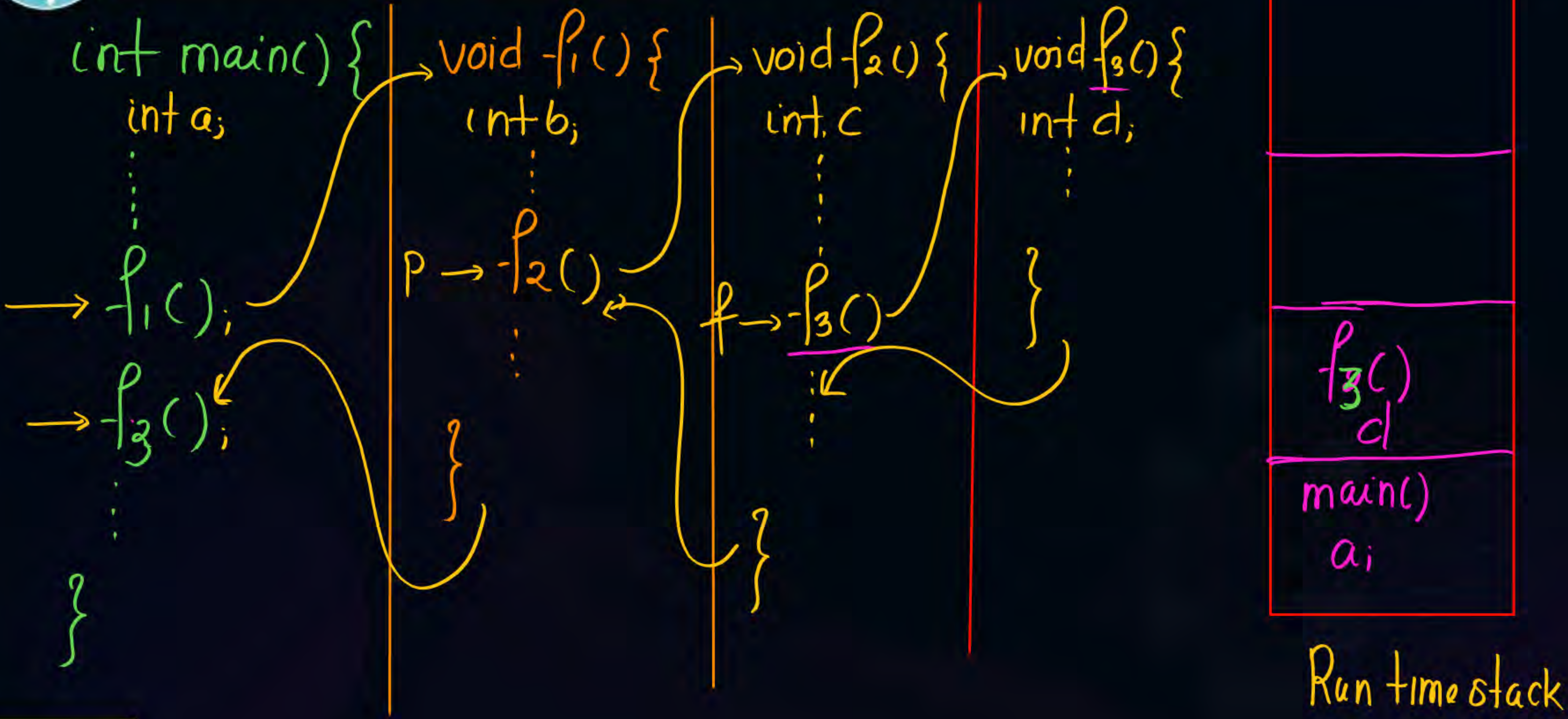
```
void f1() {
    int b;
    :
  p → f2()
    :
}
```

```
void f2() {
    int c
    :
  f → f3()
    :
}
```

```
void f3() {
    int d;
    :
}
```

Slide

Function



```
int main() {
    int a;
      :
  → f1();
  → f3();
      :
}
```

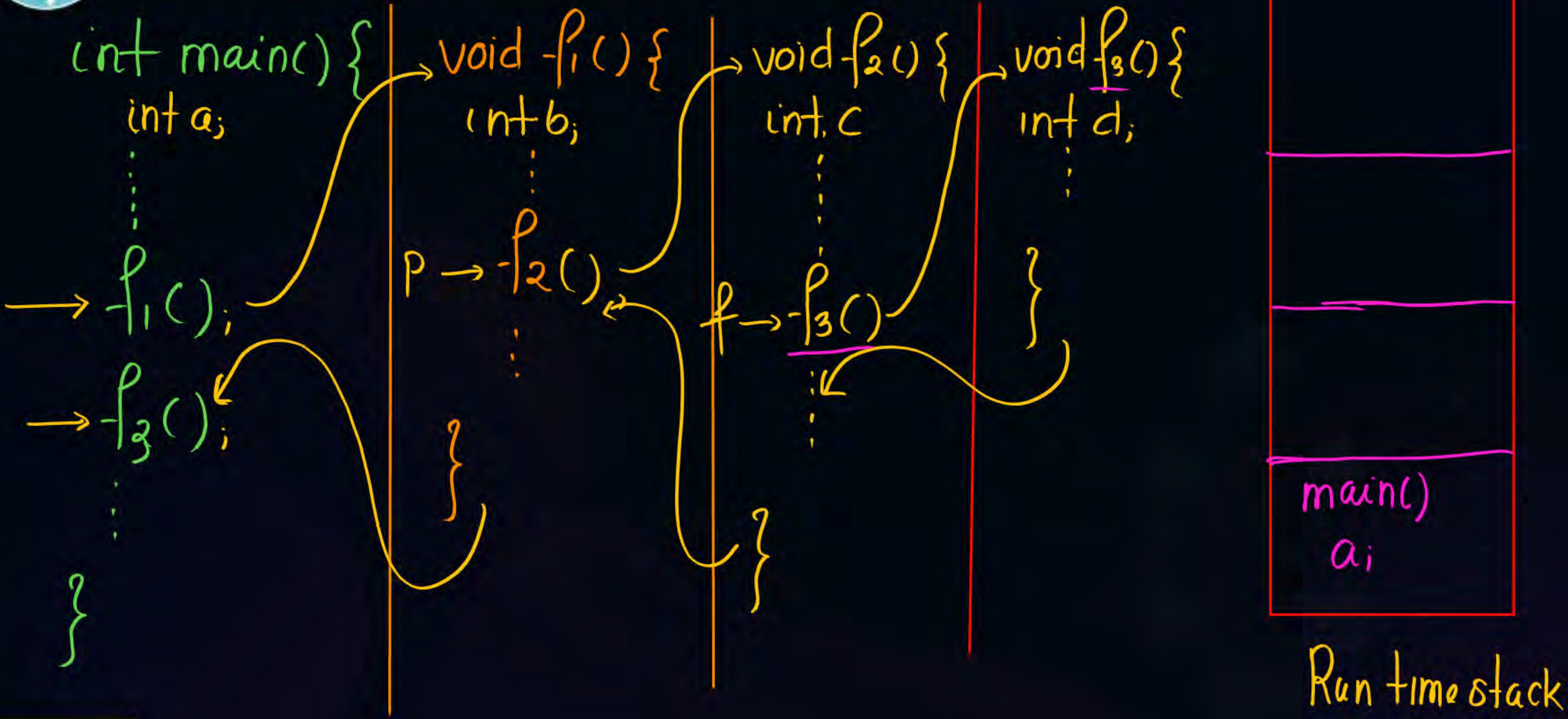
```
void f1() {
    int b;
      :
 p → f2()
      :
}
```

```
void f2() {
    int c
      :
 f → f3()
      :
}
```

```
void f3() {
    int d;
      :
}
```

Slide

Function



```
int main() {
    int a;
    :
  → f1();
  → f3();
    :
}
```

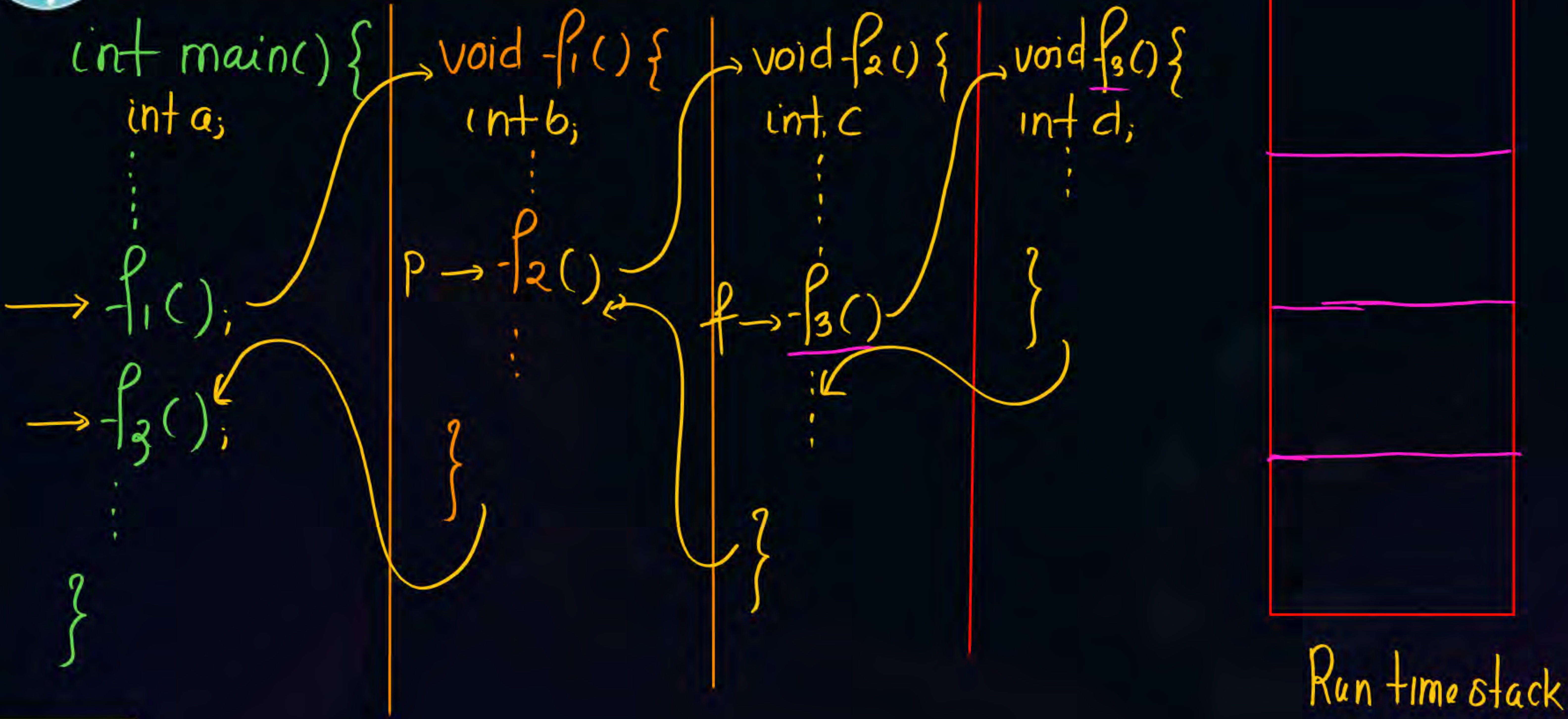
```
void f1() {
    int b;
    :
  p → f2()
    :
}
```

```
void f2() {
    int c
    :
  f → f3()
    :
}
```

```
void f3() {
    int d;
    :
}
```

Slide

Function



```
int main() {
  int a;
  ⋮
→ f1();
→ f3();
  ⋮
}
```

```
void f1() {
  int b;
  ⋮
  p → f2()
  ⋮
}
```

```
void f2() {
  int c
  ⋮
  f → f3()
  ⋮
}
```

```
void f3() {
  int d;
  ⋮
}
```

Slide

Question

#Q

No. of times '*' will be printed by the following C code is _____

```
#include<stdio.h>
void foo(int x)
{
        switch(x){
                case 1: printf("*");
                case 2: printf("*");
                case 3: printf("*");
                default: printf("*");
        }
}
int main()
{
        foo(2.5);
}
```

x = 2.5

x = 2

Definition

* * * ✓

Number of star printed is ?

(A) 2

[✓ (B) 3]

(C) 4

(D) 5

Slide

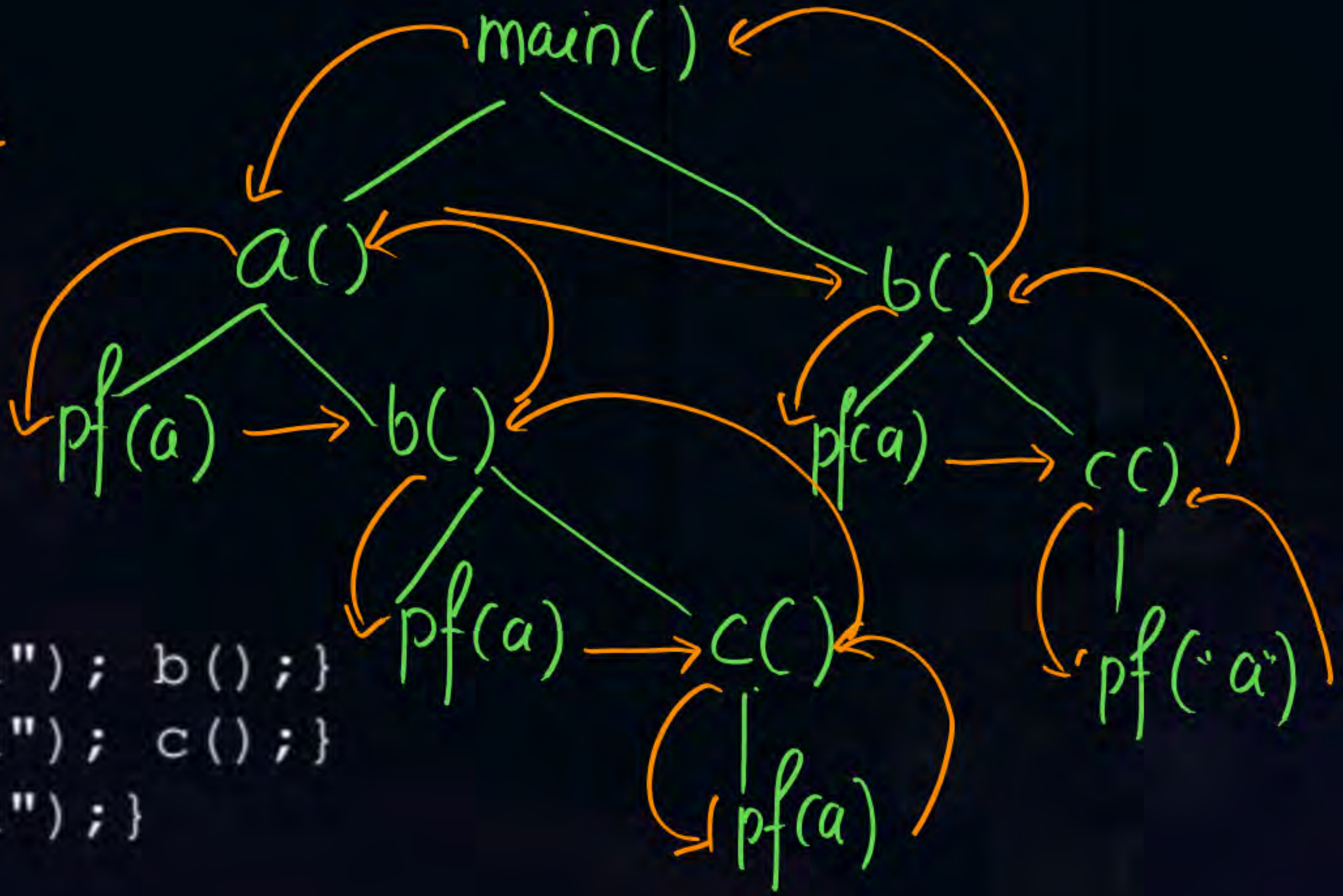
Question

a a a a a

#Q The number of character printed by the code _____

```
#include<stdio.h>
void a();
void b();
void c();
int main(){
      a();
      b();
      return 0;
}
void a(){ printf("a"); b();}
void b(){ printf("a"); c();}
void c(){ printf("a");}
```

Achvahan Tree



Slide

2 mins Summary

Topic function

Topic Activation Record

Topic practice problem

Topic

Topic

Slide

t.me/Abhisheksharmapw

THANK - YOU