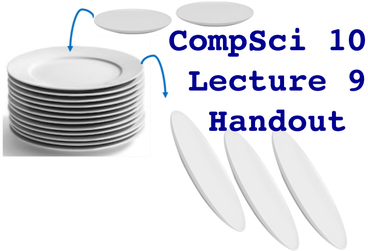
Code trace – the program stack



```
def fun_2(age):
    years = age + 10
    print("3.", years)

def fun_1(years):
    print("4.", years)
    years = 20

def main():
    years = 5
    fun_1(years)
    print("1.", years)
    fun_2(years)
    print("2.", years)

main()
```

Excercise

```
def function1():
    print("A")
    function2(3)
    print("B")

def function2(num):
    print("C")
    function4(num - 1, num - 2)
    print("D")

def function3(number):
    print("E", number)

def function4(num1, num2):
    print("F")
    function3(num1 + num2)

def main():
    print("G")
    function1()
main()
```

#the output

main() function

Code trace example

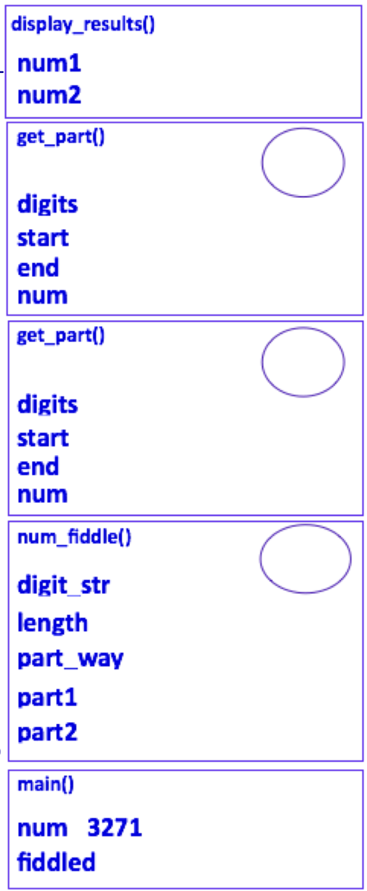
```
def get_part(digits, start, end):
    num = int(digits[start: end])
    return num

def num_fiddle(digit_str, length):
    part_way = length // 2
    part1=get_part(digit_str, 0, part_way)
    part2=get_part(digit_str,part_way,length)
    return part1 + part2

def display_results(num1, num2):
    print(num1, ", ", num2, sep = "")

def main():
    num = 3271
    fiddled=num_fiddle(str(num),len(str(num)))
    display_results(num - 5, fiddled)

main()
```



display_results()
num1
num2

get_part()
digits
start
end
num

get_part()
digits
start
end
num

num_fiddle()
digit_str
length
part_way
part1
part2

main()
num 3271
fiddled

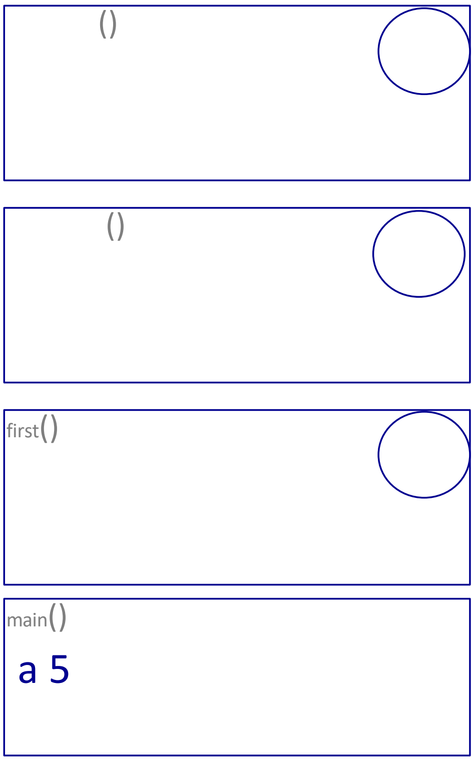
Exercise

```
def first(a):
    b = 3
    print("1.",  a)
    return second(a * b) + b

def second(a):
    print("2.", a)
    return a % 4

def main():
    a = 5
    b = first(a)
    print("3.",  b)
    b = second(b)
    print("4.",  b)


main()
```



()

()

first()

main()
a 5