

pointer operators

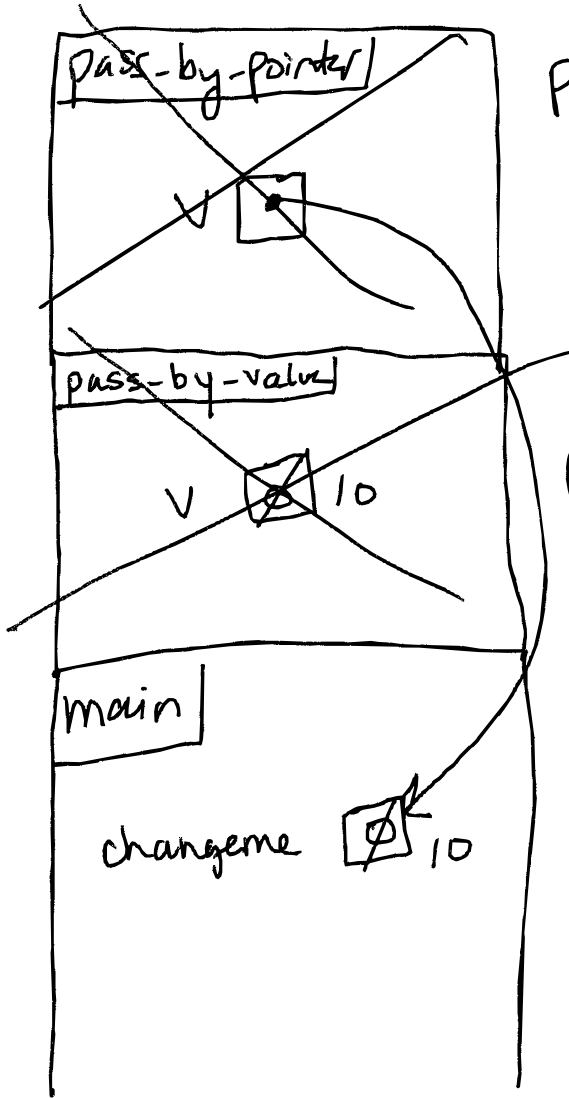
char c1 = 'a';

char* ptr; ← defining a ptr

ptr = &c1; ← address operator

char c2 = *ptr; ← dereferencing
the ptr to get the
value

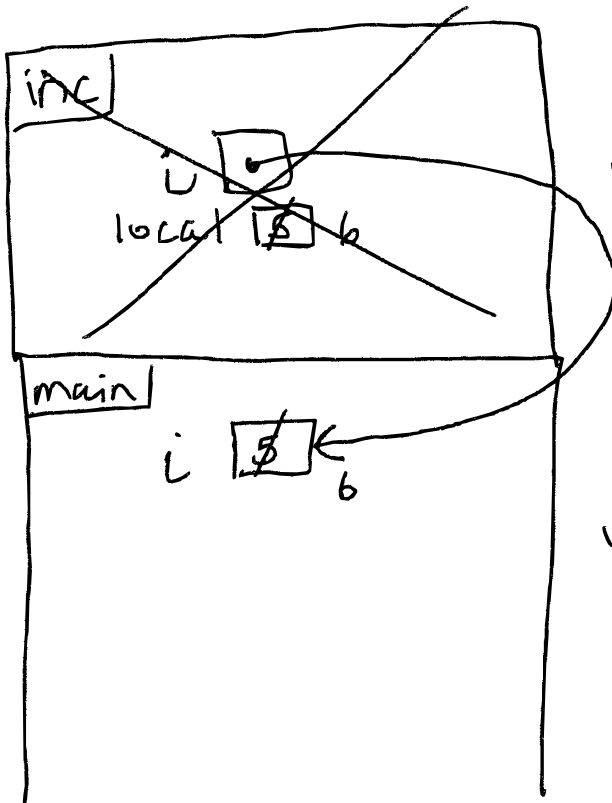
Pass by pointer: draw stack



prints : "the val
is 10"

prints : "the val
is 0"

Example: pass by pointer



prints:
"After: 6"

prints:
"Before: 5"

Struct pass by value

change Gpa

S:
gpa 3.5 3.7
newGpa: 3.7

prints
after
3.5

changeName

S: K W A M E I O
name R | V | T | H | I | O

prints
after
"Ruth"

newName R

main

student1:
name R | V | T | H | I | O
gpa 3.5

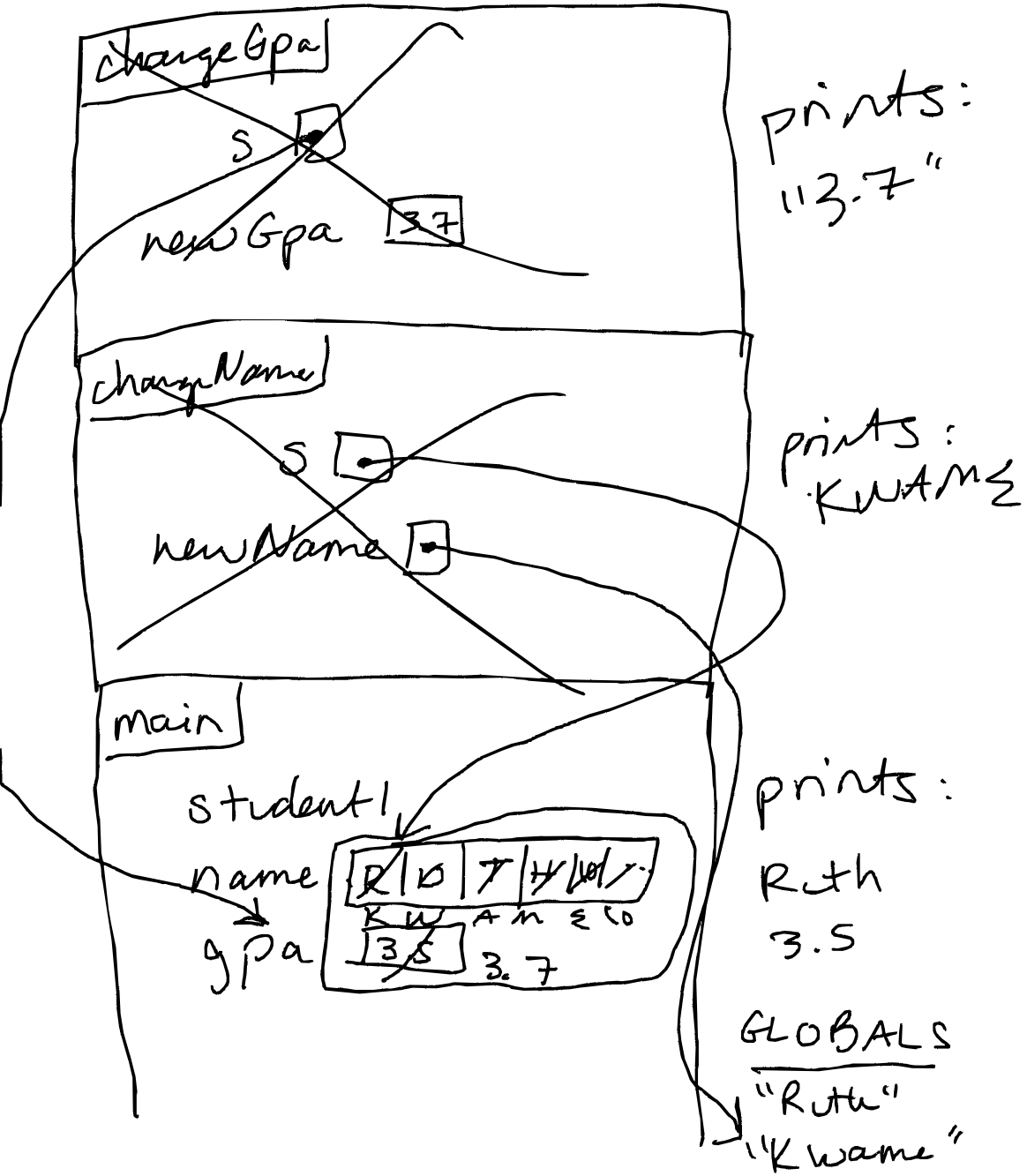
prints:

OG
"Ruth"

GLOBALS

"Ruth"
"Kwame"

Struct pass by pointer



Compiles? #1 NO

- struct passed as value
 - $\rightarrow$ is a pointer operation, should be dot
-

Compiles? #2 YES

- struct passed by ptr
- using ptr operation to update field in struct

Compiles? #3

(3 change names)

1) yes ✓

- struct passed by ptr
- → ptr op to update field

2) NO X

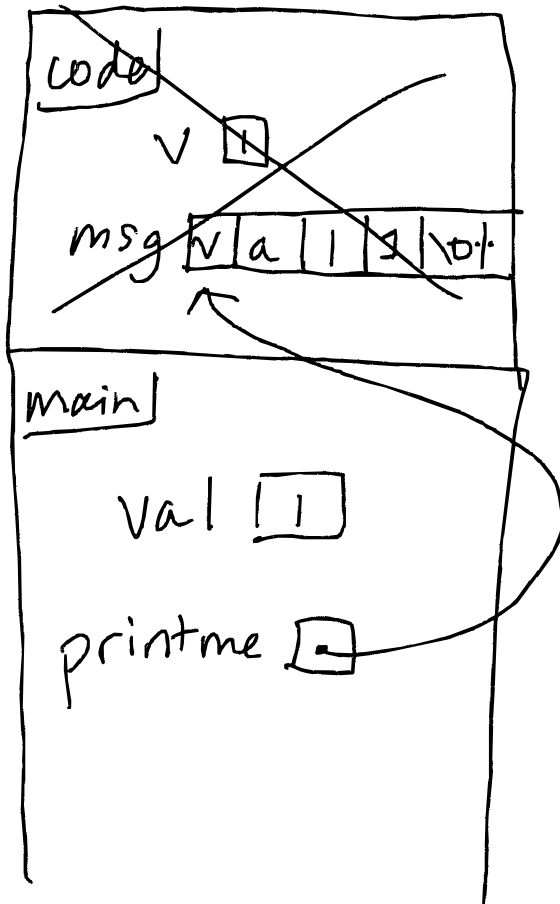
- struct passed by ptr
- can't use dot, need → operator

3) NO X

$s \rightarrow gpa = \&value$
type: float type: float* (ptr)

Return value gotchas #1

not safe x



read-only / static

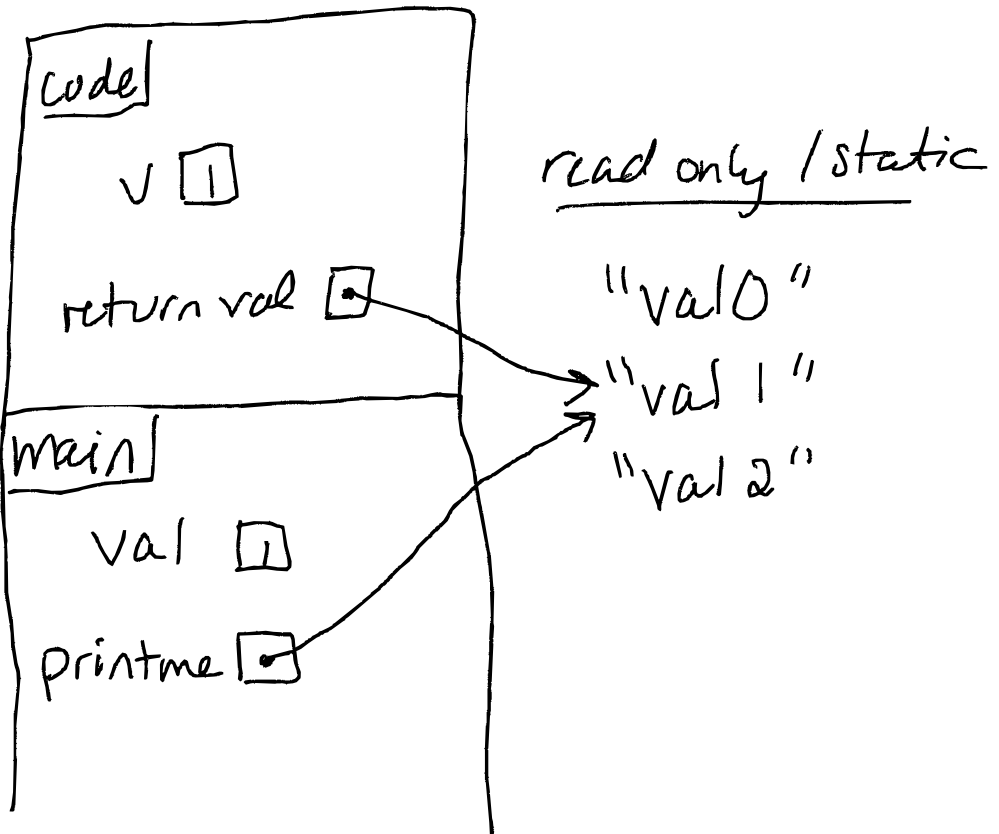
"val0"

"val1"

"val2"

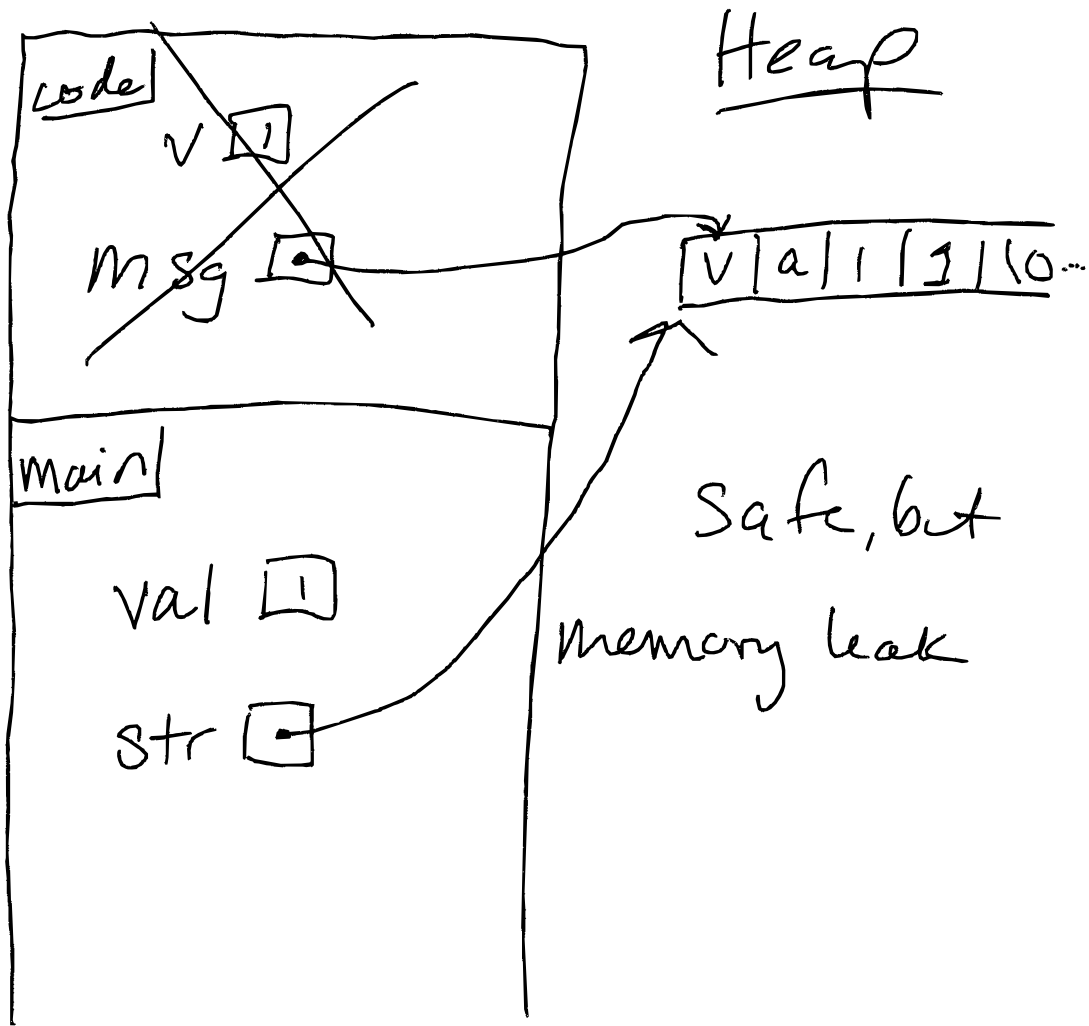
Return value gotchas #2

Safe ✓



Return value gotchas

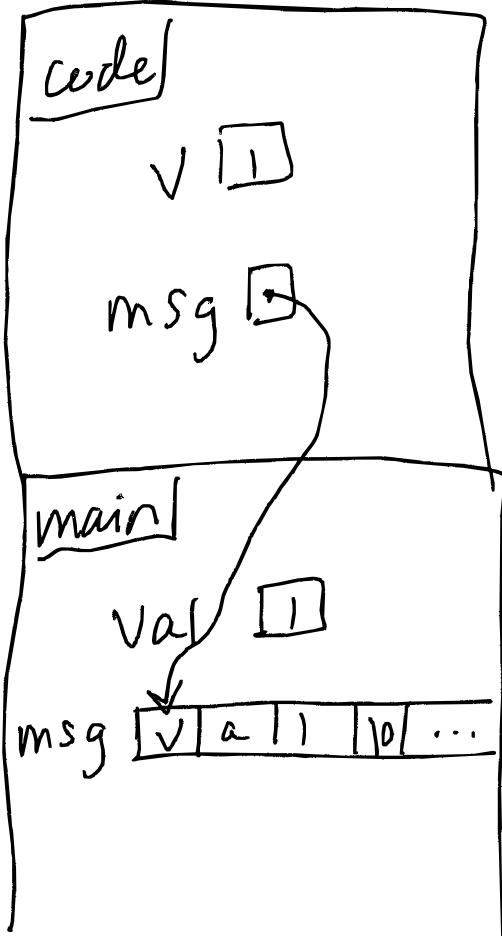
3



Return value gotchas

#4

safe ✓



passing a
ptr (b/c
array)