RAM

31/Oct/2017

← main( )

myadder (. . .)

myadder( )

bint (bye)

← ●

t_1 msecs

Overhead

- Saves all the current contents of mem/regs etc to a stack

t_2 msecs

1st

myadder

$x = a + b;$

return ●

call f Call f Call f

t

O

$t_1 = 5 \text{ msecs}$
$t_2 = 5 \text{ msecs}$

total exec. time = $T_{prog} + (3 \times 10 \text{ msecs})$

Nested func. calls

```
main() {            fA() {              fB() {

  ...                 ↓ ≡       ②          ≡
                                ③
  fA()                fB()         return;       ≡
                      ≡                 }
  fA()      ①         ≡
                                      ④      fc() {
  fA()                                          {
                      fc()
  fB()        ⑥                                  ≡
          ⑦                              ⑤
             ≡                                   ≡
  fB()      ⑧    }        ret;                    ret;
                                                 }
  ≡
  ≡
} ret;
```
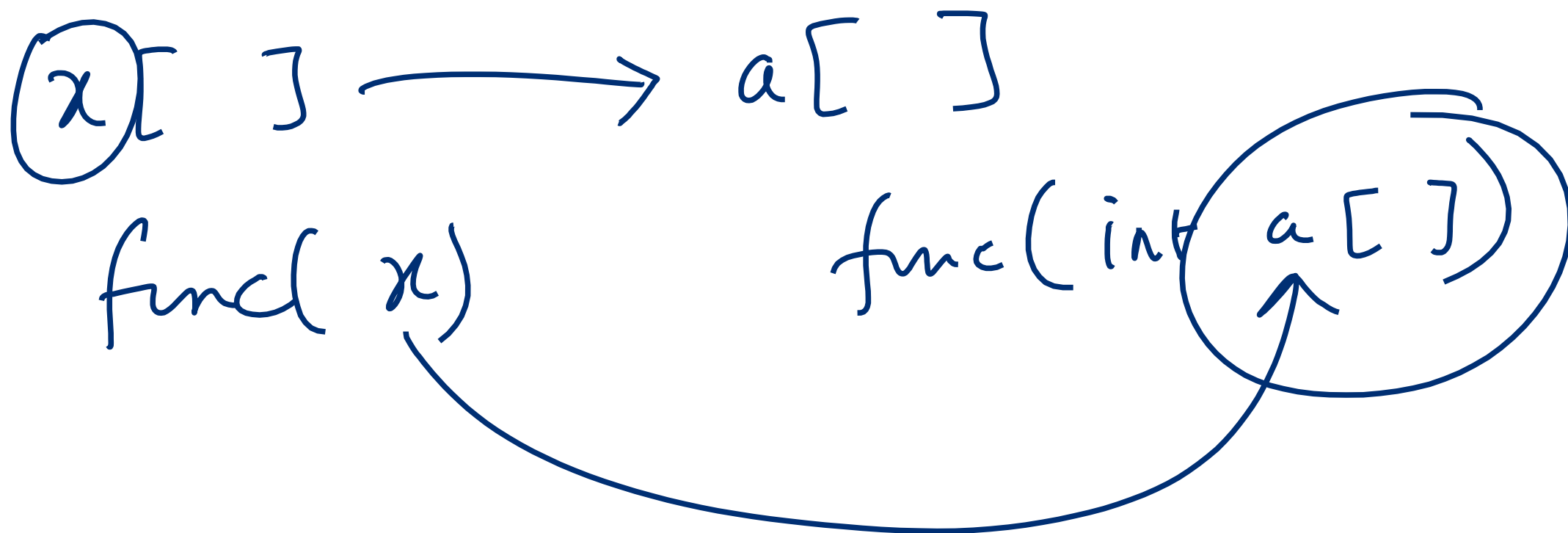
$a = x$

$func(a)$

$x = 10$

$\{$

$a = a+1$

ret

$\}$

$x[\quad] \longrightarrow a[\quad]$

$func(x)$

$func(int \; a[\;])$

array passing

int x[];

$x \longrightarrow$

$x[0]$ 1001 h

$x[1]$ 1005 h

$x[2]$ 1009 h

modify (x);

$x \longleftrightarrow$ "1001"

$x[0]$ 1001

1002

1003

1004

$x[1]$ 1005

6

7

8

$x[2]$ 1009
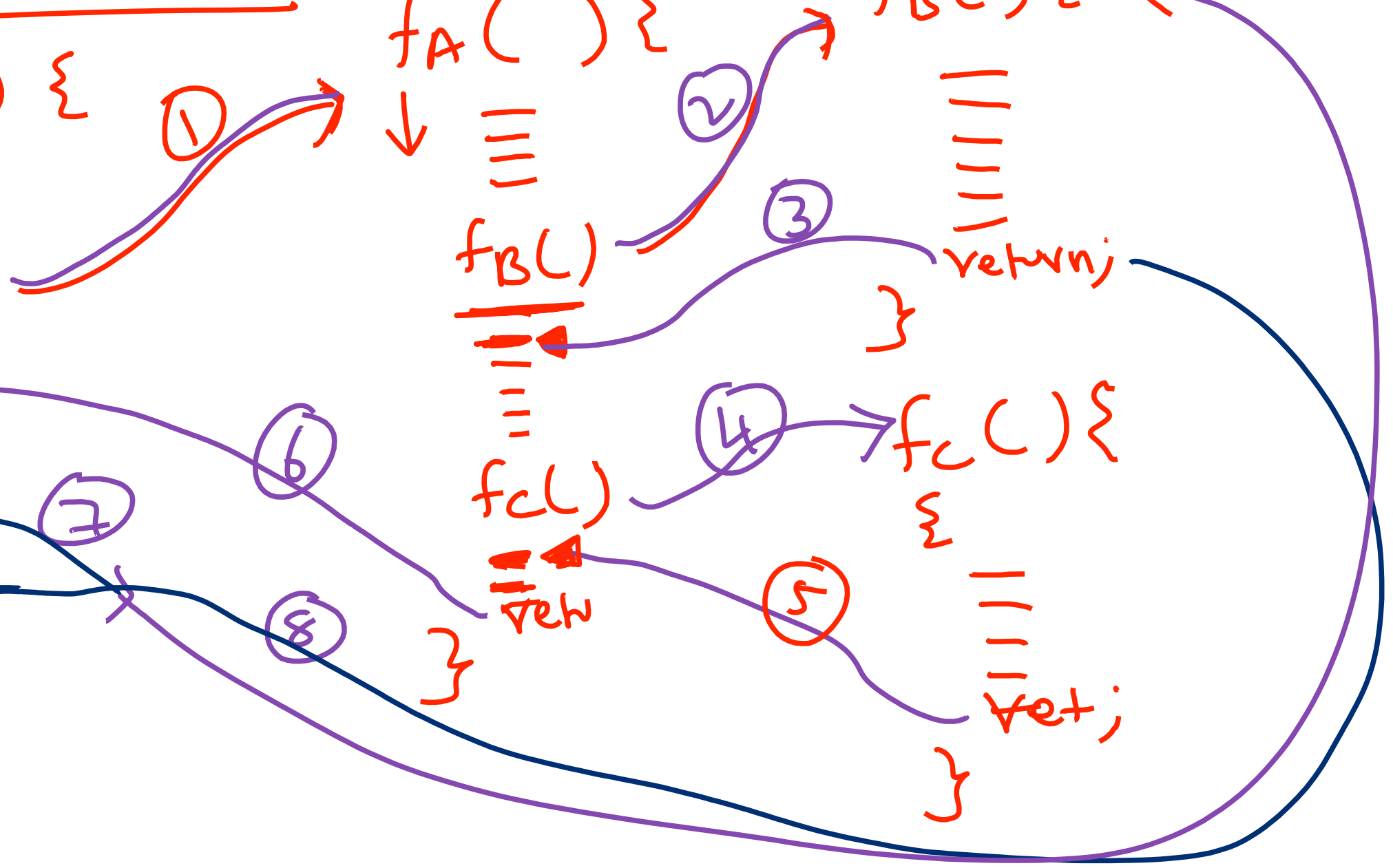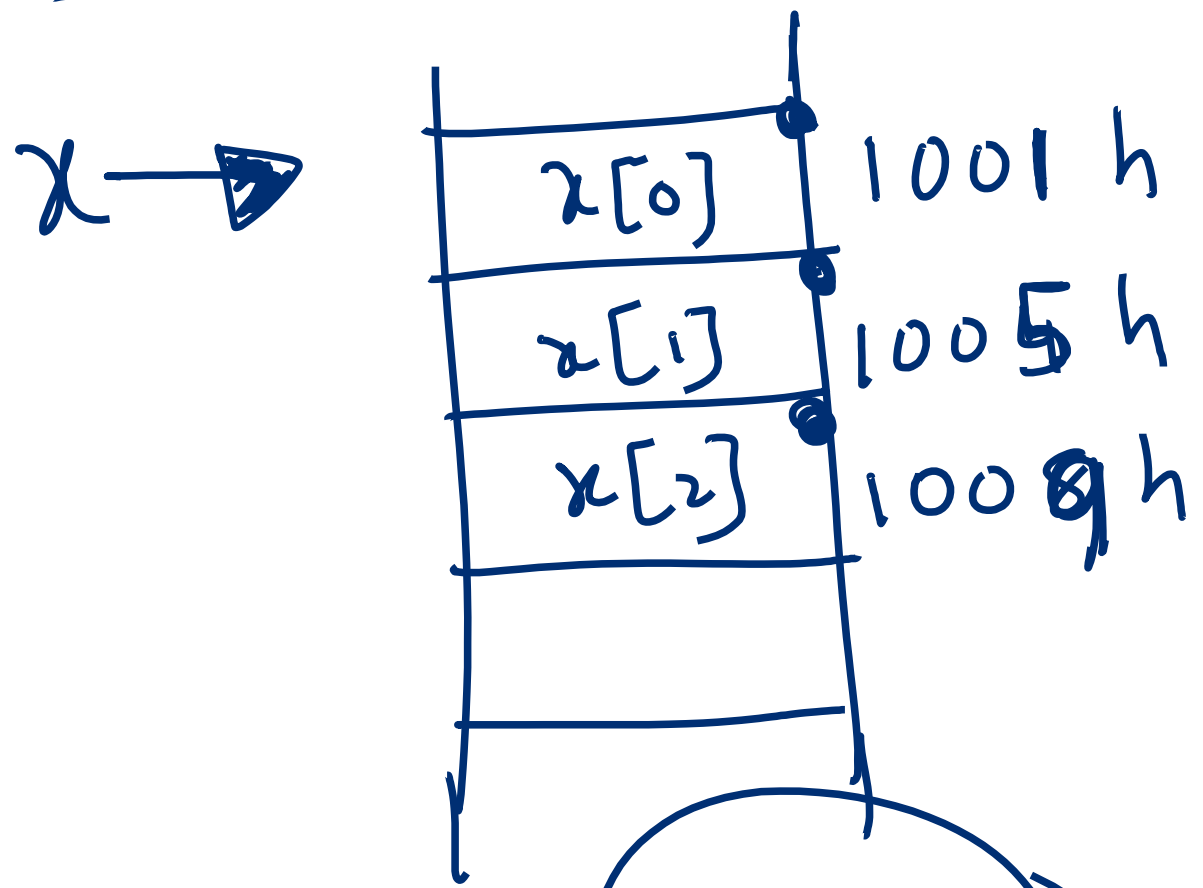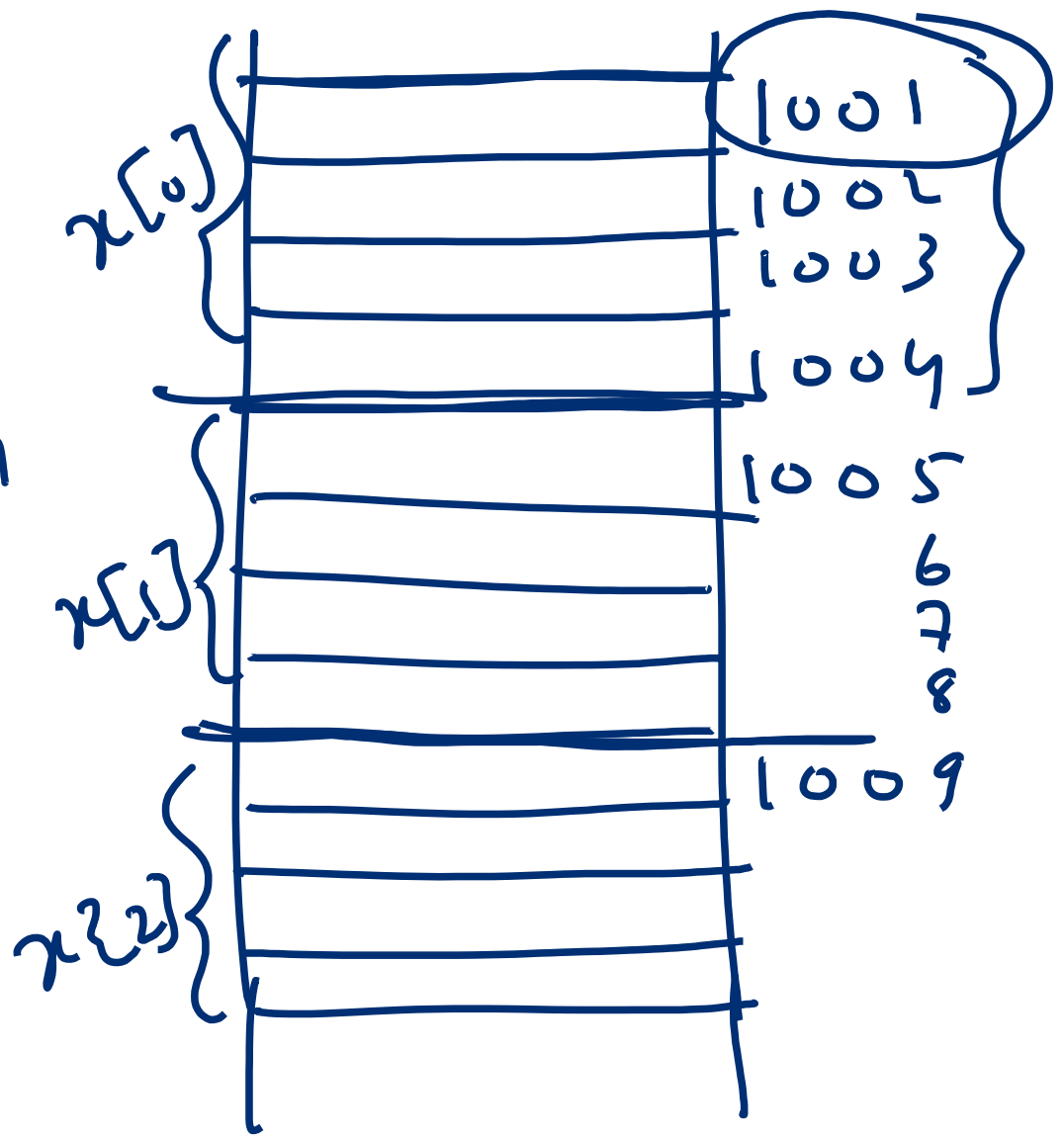
```
main( ) {
int x[3];
≡
modify(x);
≡
```

```
modify ( int A[ ] )
{
    int k; → local var
    for (k=0; k < N; k++)
    {
        A[k]
          ↑
```