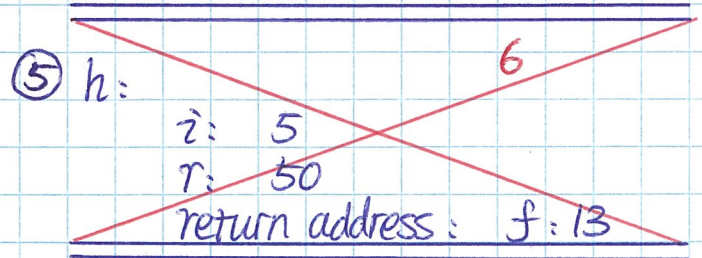
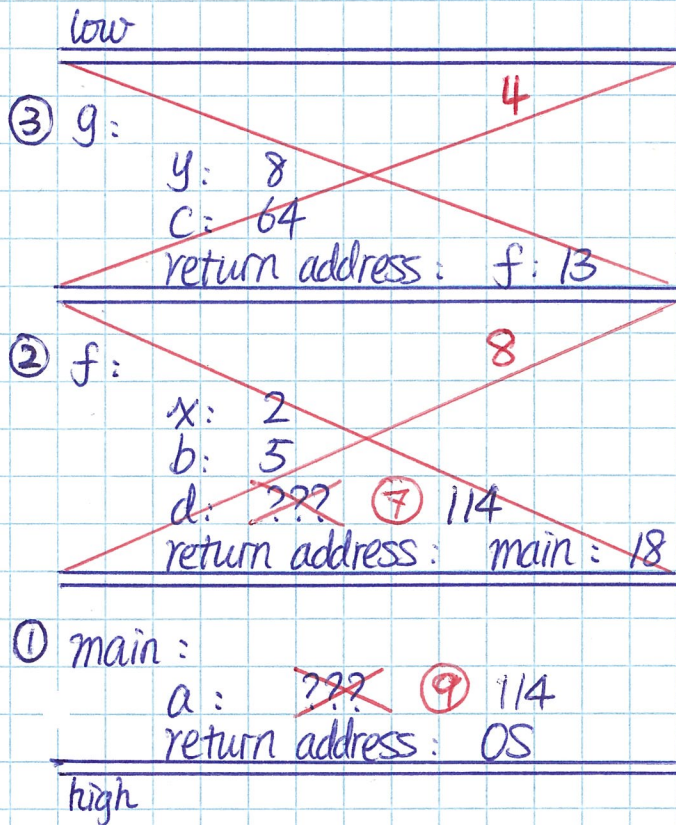
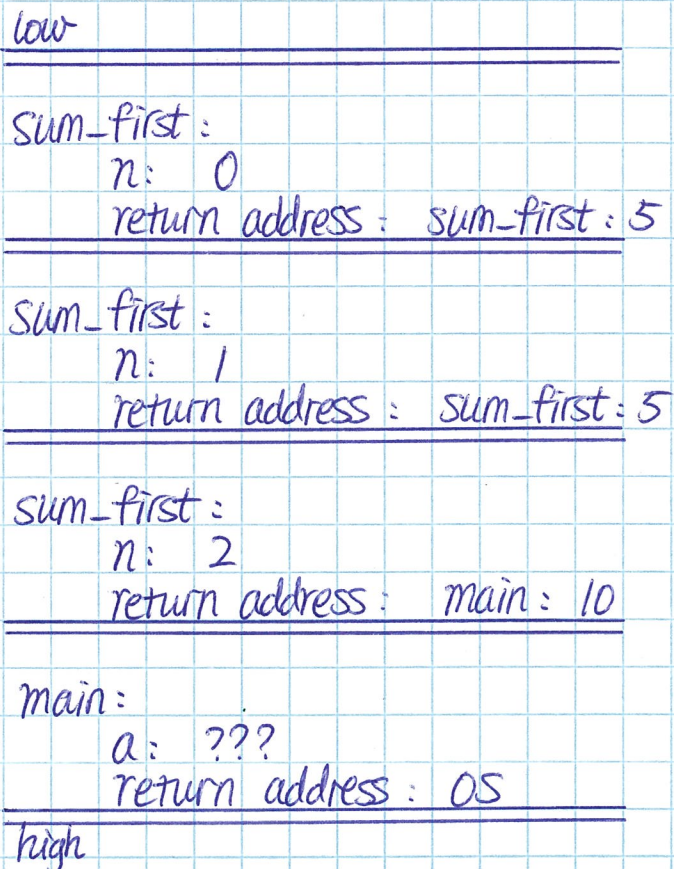


Section 4 Slides 86-87



Section 4 Slide 92



Section 4 Slide 98

```
int main(void) {
```

```
    for (int j=0; j<3; ++j) {
```

```
        int k=0;
```

```
        k = k + j;
```

```
        trace-int(k);
```

```
    }
```

```
}
```

main:

j:	0	1	2	3		
k:	0	0	0	1	0	2

return address: OS

```
1 #include "cs136.h"
2
3 int reverse_input_count(int acc) {
4     int n = read_int();
5     if (n == READ_FAIL) {
6         return acc;
7     } else {
8         int count = reverse_input_count(acc+1);
9         printf("%d\n", n + count);
10        return count;
11    }
12 }
13
14 int main(void) {
15     reverse_input_count(0);
16 }
```

Q2b. Write a program that reads in ints, and determines the count of how many numbers were read in. It prints the numbers in reverse order, adding the count to each number.

Inputs
10, 20, 30.

Outputs : 33\n 23\n 13\n

~~reverse-input-count :~~

~~count : ???~~

~~n : READ-FAIL~~

~~acc : 3~~

~~return address : reverse-input-count : 8~~

~~reverse-input-count :~~

~~count : ??? 3~~

~~n : 30~~

~~acc : 2~~

~~return address : reverse-input-count : 8~~

~~reverse-input-count :~~

~~count : ??? 3~~

~~n : 20~~

~~acc : 1~~

~~return address : reverse-input-count : 8~~

~~reverse-input-count :~~

~~count : ??? 3~~

~~n : 10~~

~~acc : 0~~

~~return address : main : 15~~

main :

return address : OS

return acc;

returns
3

returns
3

returns
3

returns
3

} return count;