Rust

```
fn foo() -> &String {

    let local_string = String::from("hello")



    & long_string



}
```

```
main
foo
local_string




"hello"


foo()
main()
```

S

h

c

C

```
char * bar() {
    char *cp = (char *) malloc(sizeof(char)*10);
    strcpy( cp, "hello" );

    return cp;
}


Main() {
    char *mcp;

    mcp = bar();

}
```

```
main
char *mcp
bar
char * cp



hello


bar()
main()
```

S

h

c

end = p + len

6

"

1☐+ 33 '\0'

p
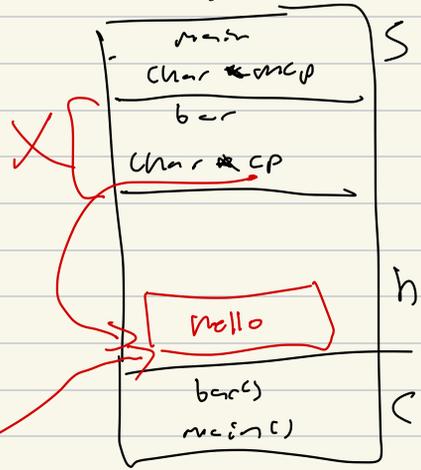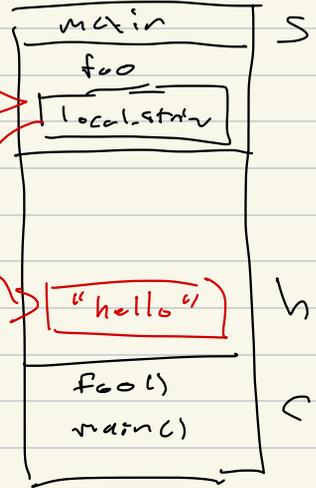
p + len

---
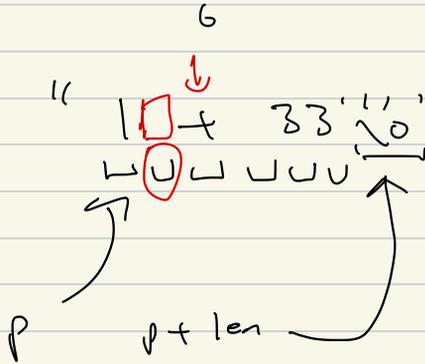
pointer   operators   and Structs

& address of

* dereference

```
struct node-st {
      int x;
      int y;
};

struct node-st node;
node.x = 1;
node.y = 2;
```

```
foo (struct node_st * np) {

    np -> x = 3      'eq     (*np).x = 3
    np -> y = 4              (*np).y = 4




}
```