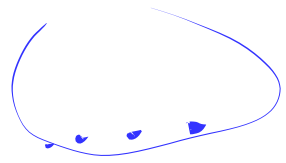




List l = 

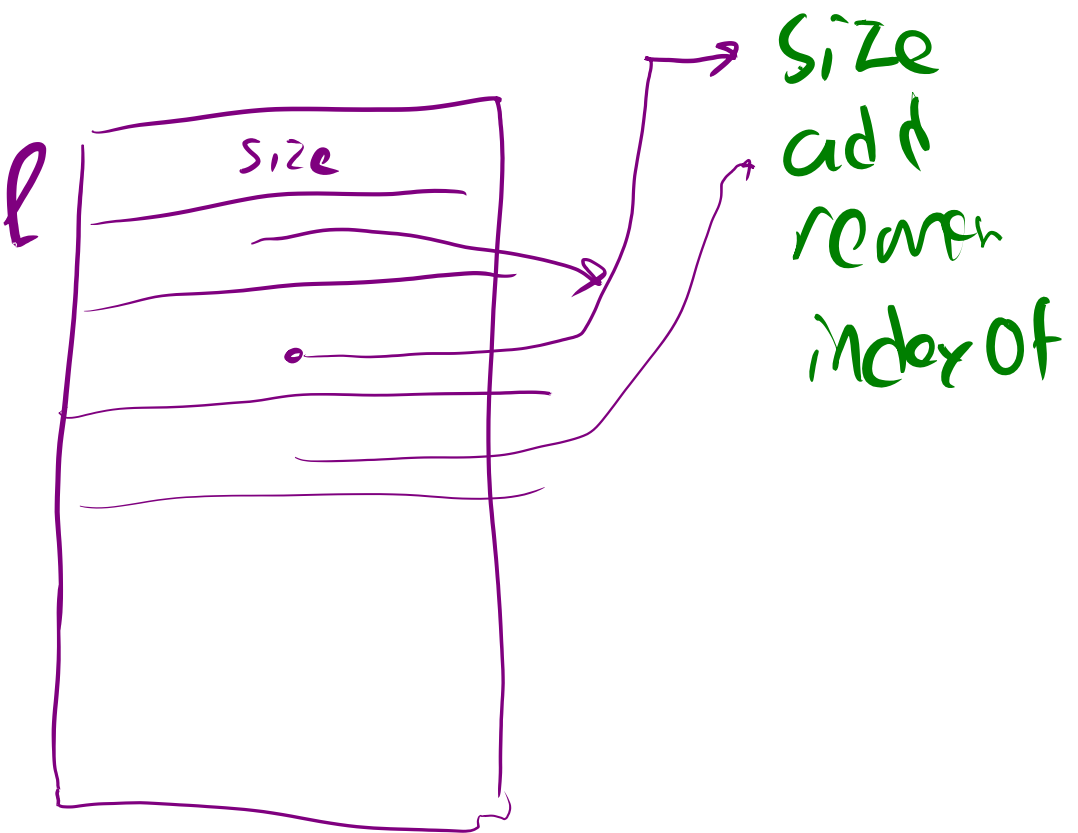
l.add(1);

List

add
if (LL) call —
if (AL) call —

Array List

Linked List



size
add
remove
index of

Extend a struct

```
struct Point {  
    int x;  
    int y;  
};
```

```
struct Point3D {  
    int x;  
    int y;  
    int z;  
};
```



```
f(Point *p) {
```

```
    Point *q = malloc(sizeof(Point))  
    *q = *p  
}
```

a. works

b. crashes

c. crashes if my code is _____

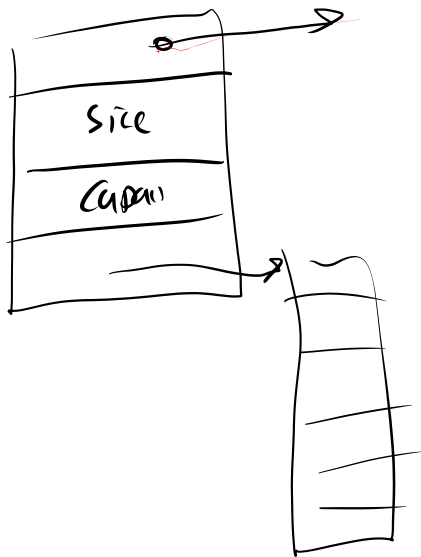
Point3D y

f((Point *) &y)

ArrList x;

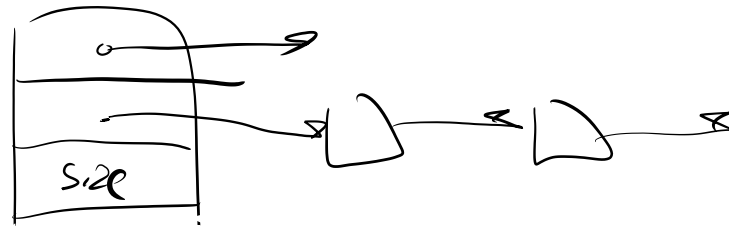
x.size()

$(*(x.vtable[0]))(AL)$

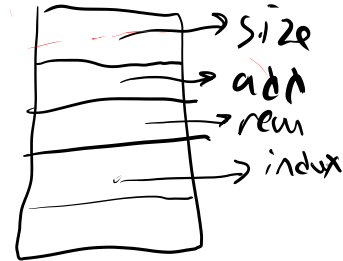


List

LL



vtable



push :-1 bp

⋮

ret

List::add:

≡
≡
≡
ret

AL::add:

≡
ret

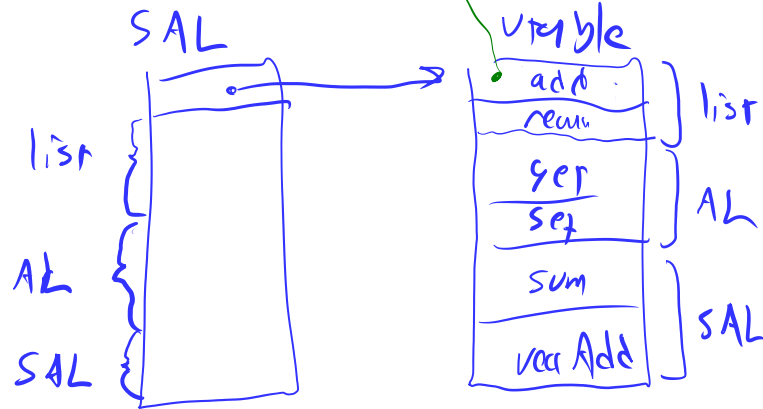
SAL: add

List
add
remove

ArrayList: List
get
set

Special ArrayList: AL
sum
vector Add

obj.vtable == ArrayList.vtable
instance of



Templates

List<string>

```
template <typename T>
    T add (T x, T y) {
        return x + y;
    }
```

```
add(3, 4)
add("hi", "there")
```

```
add g
add l
add w
add b
float-add
charb-add
```

```
T :: operator + ( )
    ⋮
```

Which lang to use

Python

Less code

good libraries

popular

C

more control

(mem)