

$r \leftarrow \sum_{i=0}^{13} \dots \rightarrow 128\text{-bit}$
 $\rightarrow 1024\text{-bit}$

$\{0, 13\} \times \{0, 13\} \times \dots$

$\sum_{i=1}^n$
 $i+1$
 $i+2$

Sage Math

$\frac{n}{n/2}$

$$K = \text{sqrt}(n)$$

for (i = 3 ; $i < \text{sqrt}(n)$; $i = i + 2$)

The diagram shows a for loop with a box around the condition $i < \text{sqrt}(n)$. An arrow labeled 'K' points to this condition. Another arrow labeled 'Linear' points to the loop body. A third arrow points from the loop body to the increment $i = i + 2$.

Complex

OBJECT

Structure

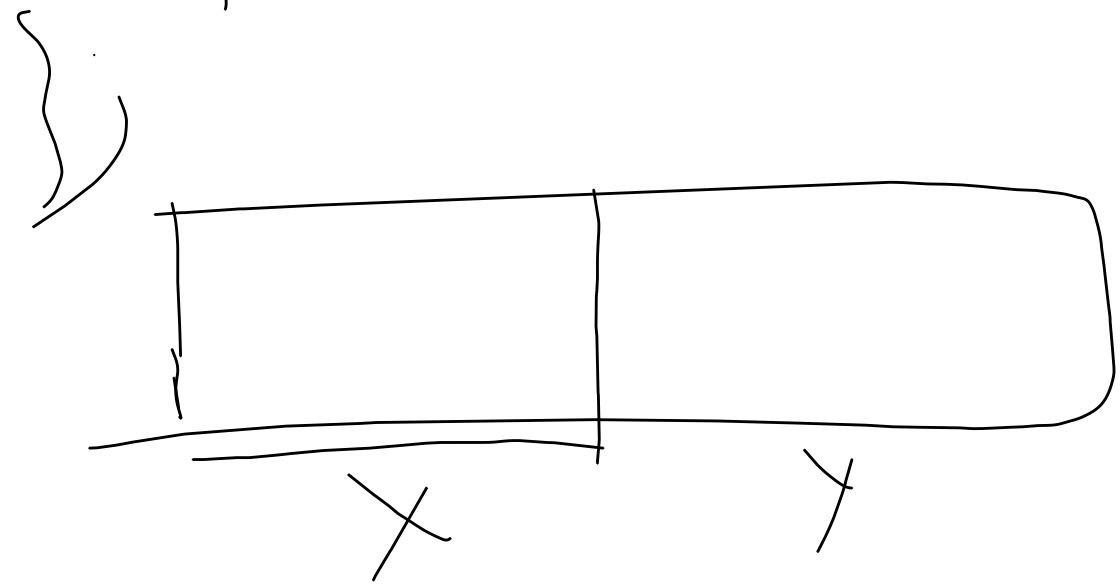
→ int → a, b
 → float → x, y
 char s

struct Complex
 float x
 float y

Complex → (x, y)

struct Complex ≅

2x



✓

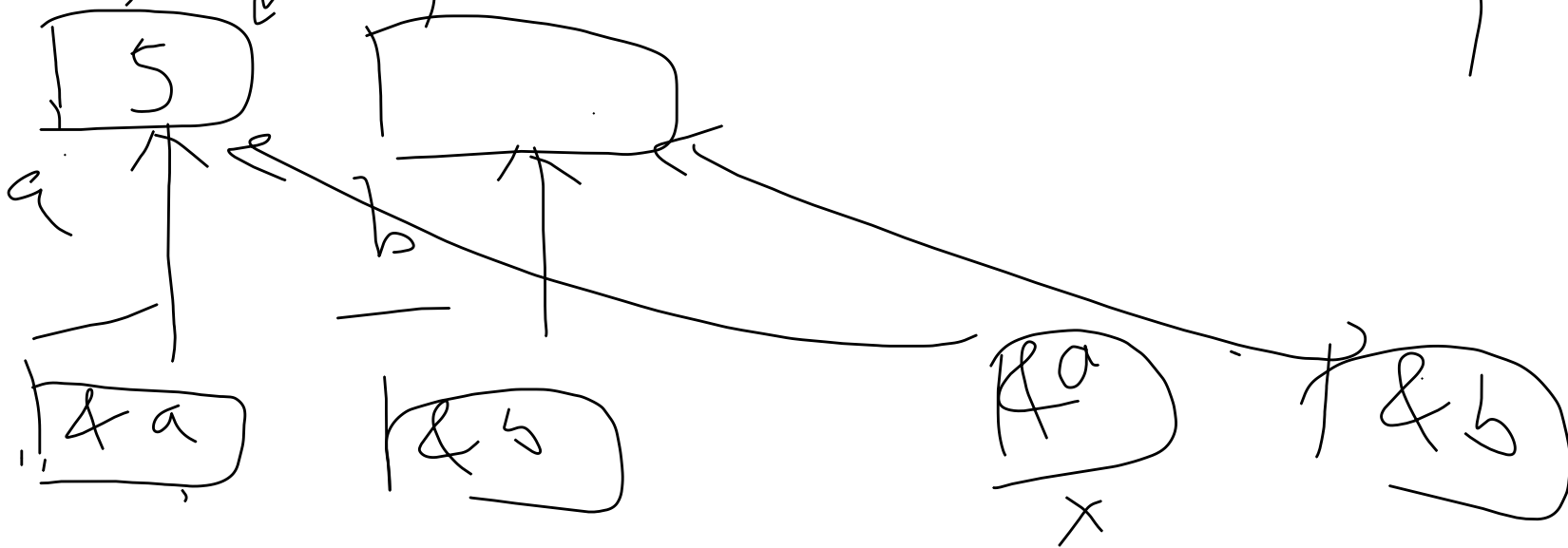
Swap (int *a, int *b)

c = a
a = b
b = c

Swap (int *x, int *y)
int c; int *x; int *y

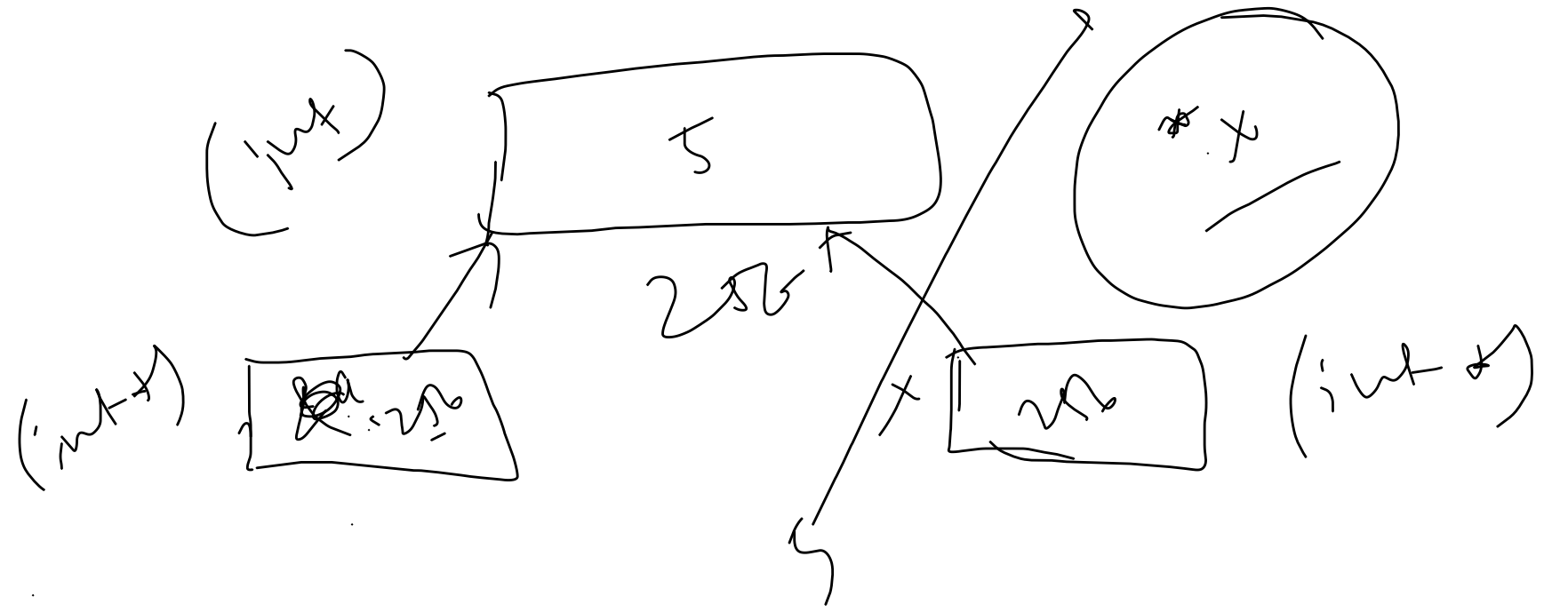
*x = *x
*y = *y
*y = c

```
main()
{
  int a, b; a = 5; b = 10;
  swap(a, b);
  print(a, b);
}
```



]

```
int a;  
swap(a, b)  
// a, b  
Swap(int *x, int *y);
```



Struct Complex {

float x;

float y;

}

float x, y;

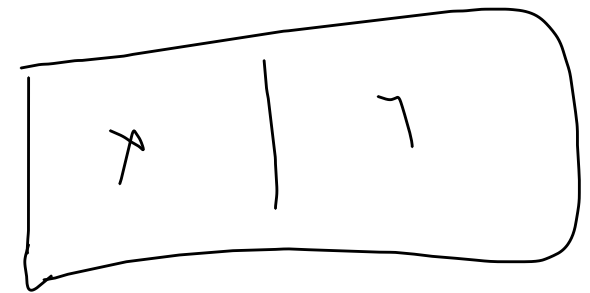
Struct Complex z1, z2;

$$z1 \cdot x = 1.0$$

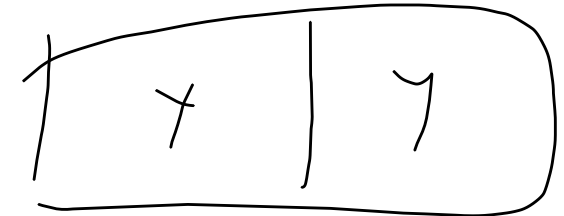
$$z1 \cdot y = 2.0$$

$$z2 \cdot x = -1.0$$

$$z2 \cdot y = -2.0$$



z1



z2

void

print_complex (str^{ull} complex z) {

printf ("%f + i %f", z.x, z.y) }

}

a + ib
(a, b)

```
typedef struct Complex Complex;
}
}
print-Complex(struct Complex)
}
int main()
}

typedef unsigned int uint;
```


Complex

Add_complex (Complex z1, Comp) { z2.

Complex z3;

$$z3.x = z1.x + z2.x;$$

$$z3.y = z1.y + z2.y;$$

Return z3;

}

Student }

Row

int subj1;
int subj2;
char grade;



$0 \leq \leq 100$

char

ComputeGrade (Student st)
 $\frac{st.subj1 + st.subj2}{2}$

$st \rightarrow subj1$

main {
Student s1, s2;

s1.grade = ComputeGrade (s1)

/