$r \leftarrow \{0,1\}^\lambda$ → 128-bit
1024-bit

$\{0,1\} \times \{0,1\} \times \ldots$

n

$n/2$

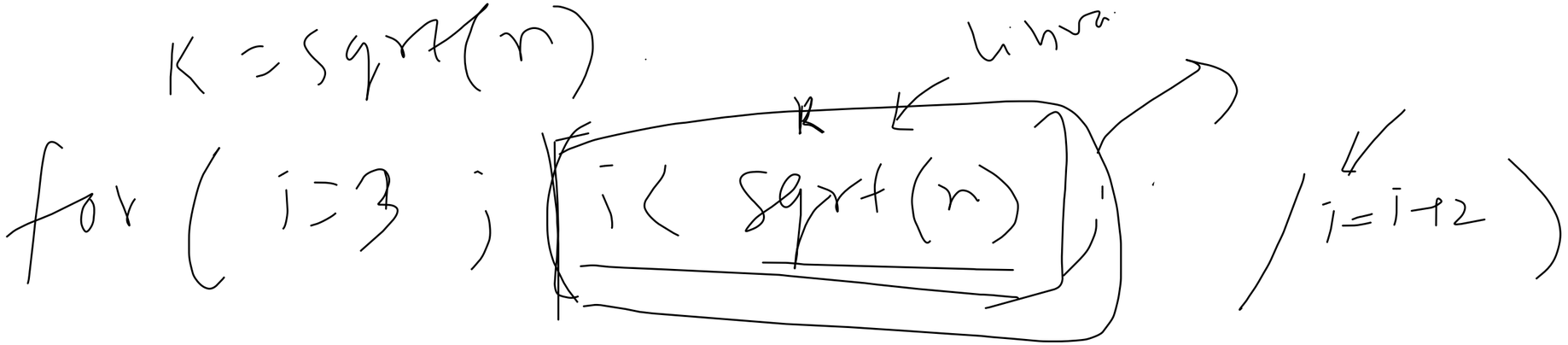
sqrt(n)

i+1

i=i+2

SageMath

K = sqrt(n)



OBJECT

Structure.

Complex

→ int → a, b;

→ float → x, y;

char s

struct Complex {

→ float x

float y.

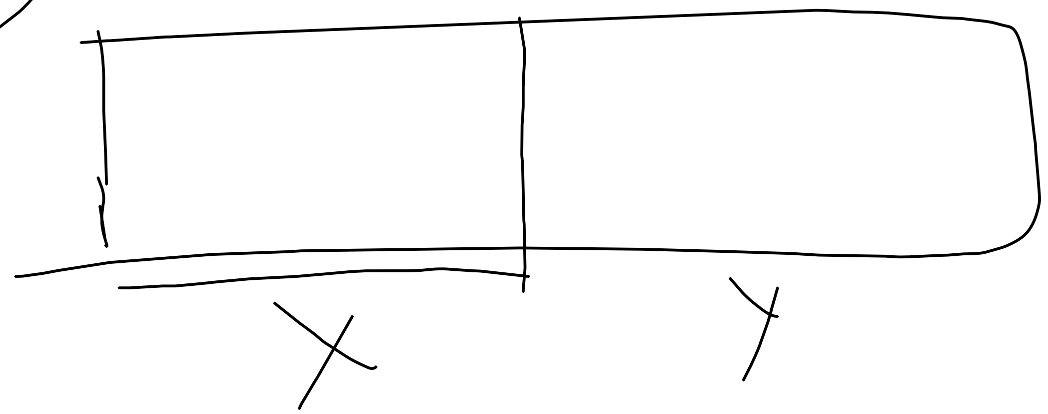
};

Complex → (x, y)

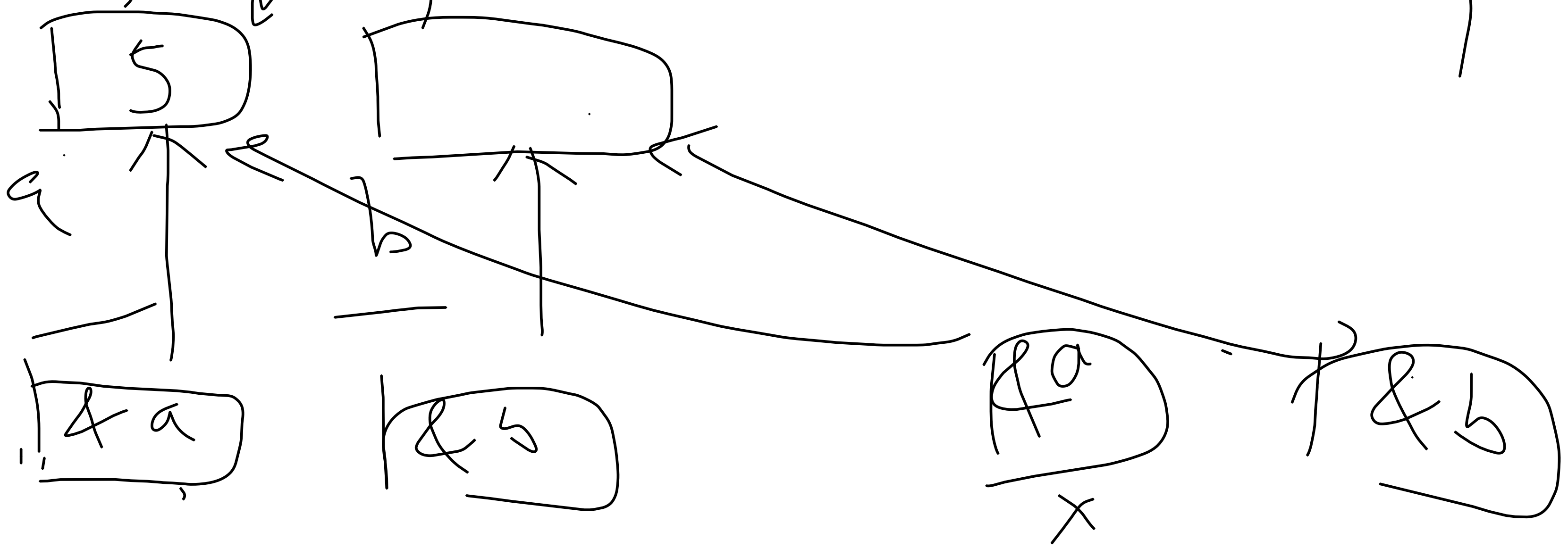
struct Complex z.

z.x

✓



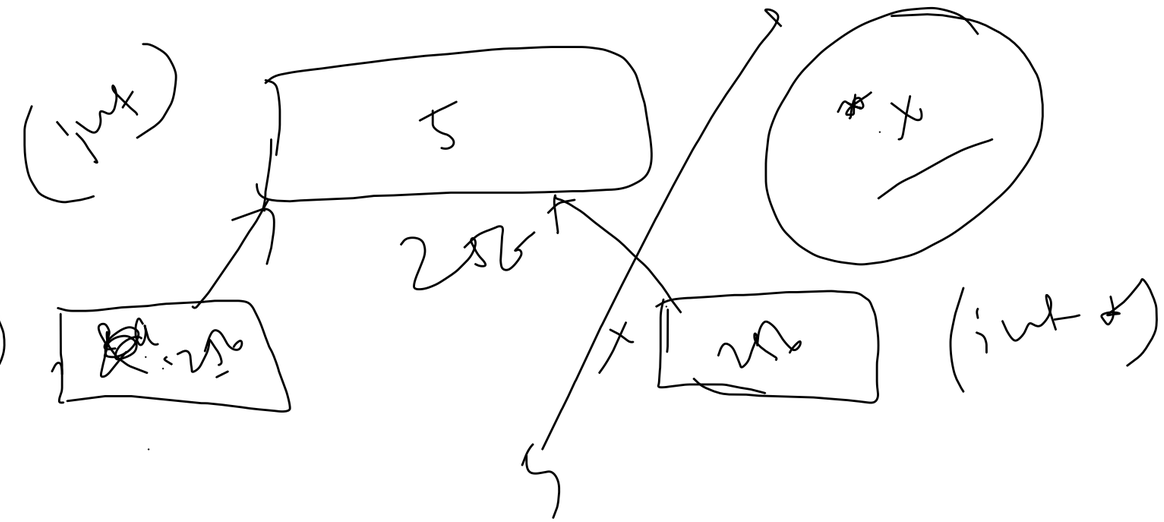
```
Swap(int*a , b){
    c = a
    a = b
    b = c
}

Swap(int *x, int *y){
    int c;
    c = *x
    *x = *y
    *y = c
}

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



int a;

swap(&a, b)

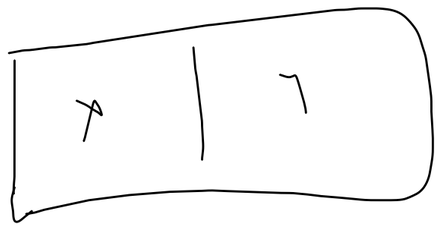
swap(int* x, int* y);



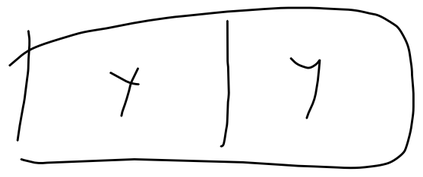
```
struct Complex {
    float x;
    float y;
};
```

```
struct Complex z1, z2;
```

z1.x = 1.0
z1.y = 2.0
z2.x = -1.0
z2.y = -2.0



z1



z2

void print_complex(struct Complex z){

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

}

$\hat{a}+ib$

(a, b)

typedef struct Complex Complex;

struct Complex {

}

print_Complex(struct Complex)

}

int main() {

}

Complex 2

int

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 {

Roll

int Subj1 ;

int Subj2 ;

char Grade ;

}

$0 \leq \leq 100$

char ComputeGrade (Student *St)

{

St.Subj1 + St.Subj2

St -> Subj1

}

main {

Student St1 St2 ;

→ St1.grade = ComputeGrade (&St1)