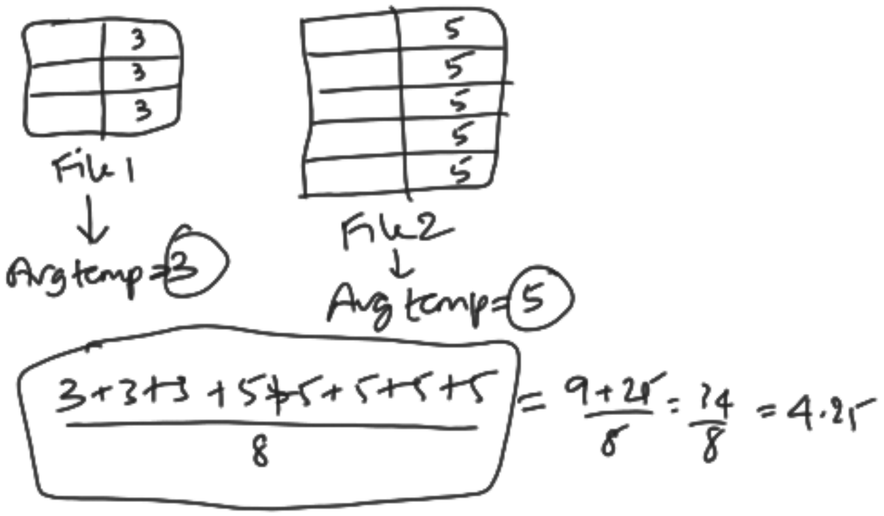


$$\frac{3+3+3+5+5+5+5+5}{8} = \frac{9+25}{8} = \frac{34}{8} = 4.25$$

invalid temp → -9999
invalid humidity → "N/A"

File 1

	H	T
	20	75
	24	-9999
	N/A	65

Avg temp = (75+65)/2 = 70
Avg humidity = (20+24)/2 = 22

File 2

	H	T
	20	-9999
	21	15
	22	16
	N/A	17

Avg temp = (15+16+17)/3 = 16
Avg hum = (20+21+22)/3 = 21

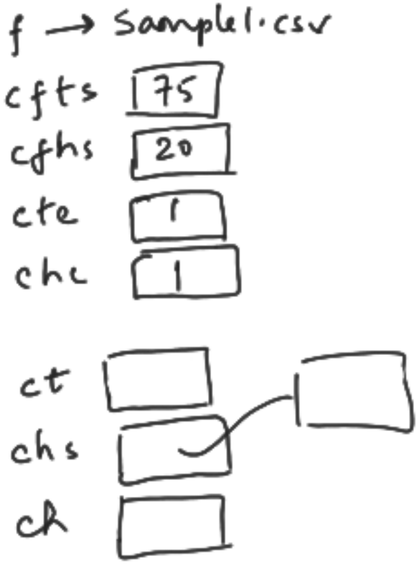
Avg temp = (75+65+15+16+17)/5 = 188/5 = 37.6
Avg hum = (20+24+20+21+22)/5 = 107/5 = 21.4

=
f . File f : dr. sF) {
double totTemp
for (

File : Sample1.csv, Avg temp: 70, Avg hum: 22
File : sample2.csv, Avg temp: 16, Avg hum : 21

Summary: Avg temp : 37.6, Avg hum : 21.4

```
DR dr =
double finalTotalTemp = 0.0
int finalNumTempEntries = 0;
for (File f : dr.sf()) {
    double totTemp = 0;
    int numTempEntries = 0;
    for (record : power) {
        finalTotalTemp += currTemp
        totTemp += currTemp;
        numTempEntries += 1;
    }   finalNumTempEntries += 1;
    double avgTemp = totTemp/numTe
    print(...)
}
double totalAvg = finalTotalTemp / finalNumTempEntries
```


f → sample1.csv
cfts
75
cfhs
20
cte
1
chc
1
ct
chs
ch

```
getRank(year, name, gender){

    String fileName = getFileName(year);
    FR fr = new FR(filename);
}

getName(year...){
    String filename = getFileName(year);
    ...
```

```
String getFileName(int year){
  //return "testing/yob" + year +
                    "short.csv";
  return "data/yob" + year + ".csv";
}
```

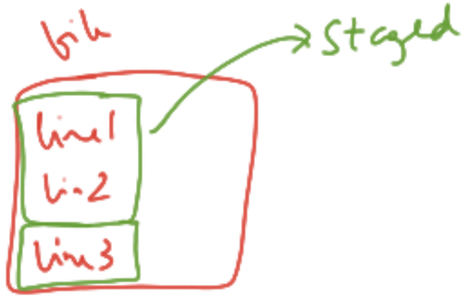
f("hello", 2) → "eo"

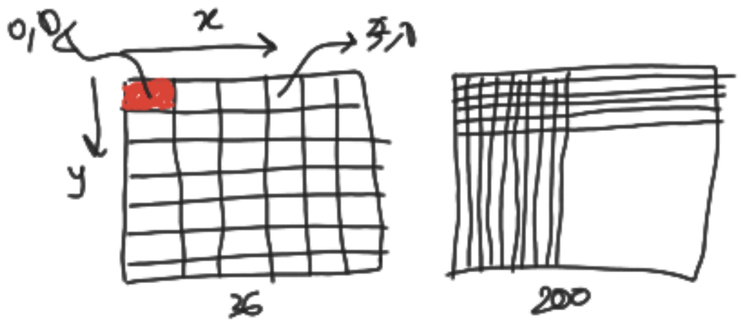
f("hello", 1) → "e"

f("hello", 3) → "invalid"

f("hello", 0) → ""

```
print("hello '")    // 'hello '
print('hello')      // 'hello'
```

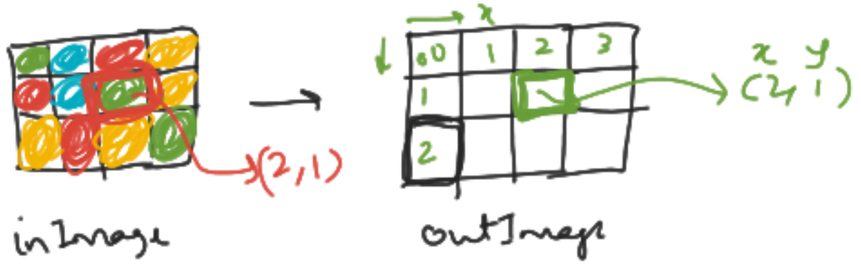




Color → Red 255 0-255
Green 0 0-255
Blue 0 0-255

Red 0
Green 0
Blue 0
} Black

Red 255
Green 255




```
for (Pixel outPixel : outImage.getPixels()){
    int x = outPixel.getX(); → 2
    int y = outPixel.getY(); → 1

    Pixel inPixel
      = inImage.getPixel(x, y);

0 ← int red = inPixel.getRed();
255 ← int green = inPixel.getGreen();
0 ← int blue = inPixel.getBlue();

    int gray = (red + blue + green)/3;
      ↙
    85

    outPixel.setRed(gray)
    outPixel.setGreen(gray)
    outPixel.setBlue(gray);
}
```

```
int width = inImage.getWidth();
int height = inImage.getHeight();
IR outImage = new IR(width,
                     height);

for(Pixel inPixel : inImage.pixels()){
    int x = inPixel.getX();
    int y = inPixel.getY();

    Pixel outPixel
        = outImage.getPixel(x, y);

    int red = inPixel.getRed();
    int green = ...
    int blue = ...
    int gray = (red + green + blue)/3;

    outPixel.setRed(gray);
    outPixel.setGreen(gray);
    outPixel.setBlue(gray);
}
```

```
for(Pixel outPixel : outImage.pixels()){
    int x = outPixel.getX();
    int y = outPixel.getY();

    Pixel inPixel =
        inImage.getPixel(x,y);
    ⋮
```


File f
name
path
inImage
name
path
image
copy-1.jpg
1.jpg
.../Downloads/
images
grayImage
name
path
image
gray-1.jpg
../DukeHW/
HIPP

[1, 2, 3, 1] len

(1,2)
(1,3)
(1,1)

(2,3)
(2,1)

(3,1)

```
for (i=0; i<len; i++) {
    for (j=i+1; j<len; j++) {
        if (input[i] == input[j]) {
            return true;
        }
    }
}
return false;
```

abcd dddd

for loop

char → 'd'

anagramWordCharCount = 4

originalWordCharCount = 1

org ana

aabcccdef aabccc

a →

anaCount = 2

orgCount = 2

a →

b →

ac = 1

oc = 1

c →

ac = 3

oc = 3

[3,7,4,3,2,7,5,3]

↓

[2,3,3,3,4,5,7,7]

I AM

encry

""

"Z"

"Z "

"Z R"

"Z RD"

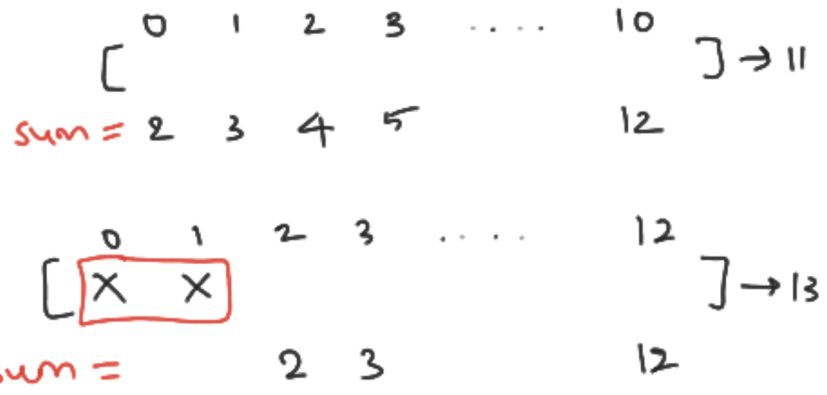
Alphabet.: ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
GHIJKLMNOPQRSTUVWXYZABCDEFghijklmnopqrstuvwxyzabcdef

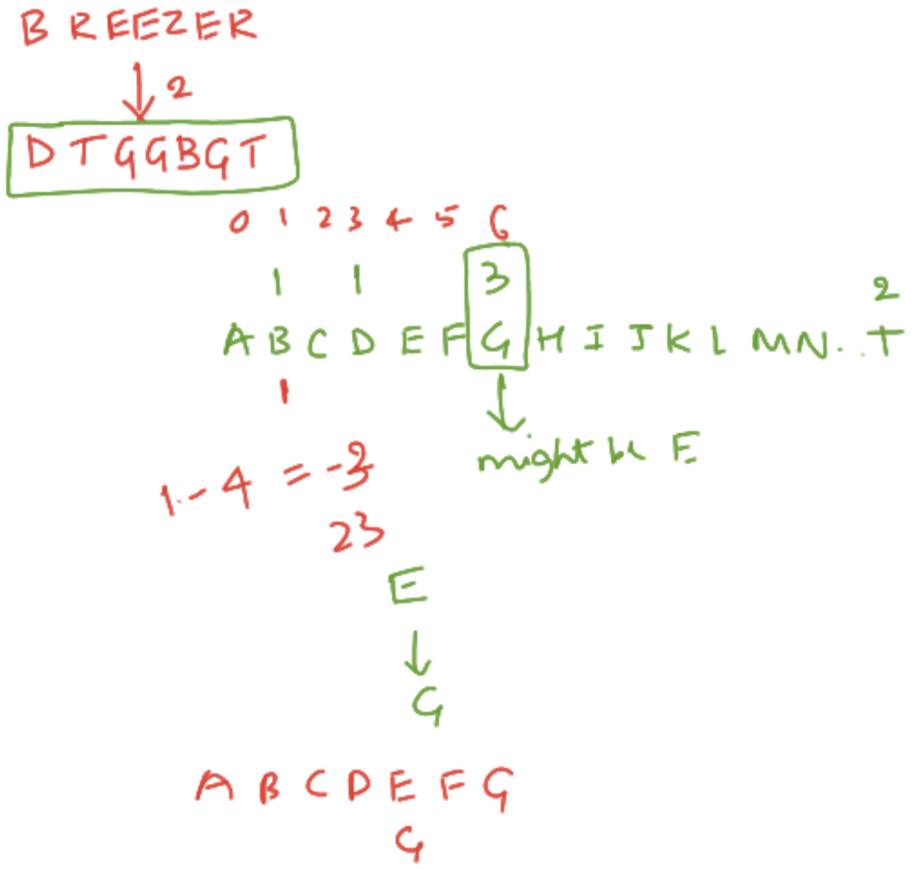
HELLO
↓ encrypt with 6
NKRRU
↓ encrypt with key (20)
HELLO

20 → 26-key

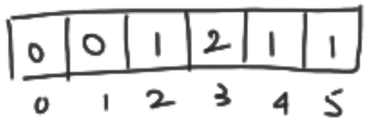
Hello World!
↓ enc(6)
Nk

HELLO WORLD!
NKRRU CUXRJ!


0 1 2 3 10
] → 11
sum = 2 3 4 5 12
0 1 2 3 12
X X
] → 13
sum = 2 3 12

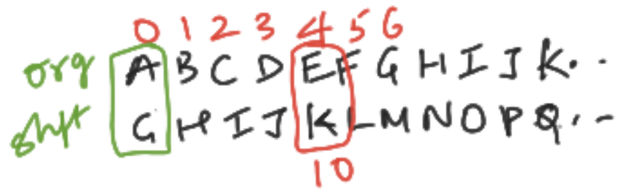

BREEZER
2
DTGGBGT
0 1 2 3 4 5 6
1 1 3
A B C D E F G H I J K L M N . . T
1 2
might be E
1-4 = -3
23
E
G
A B C D E F G
G

hello, who is this guy
5 3 2 4 3

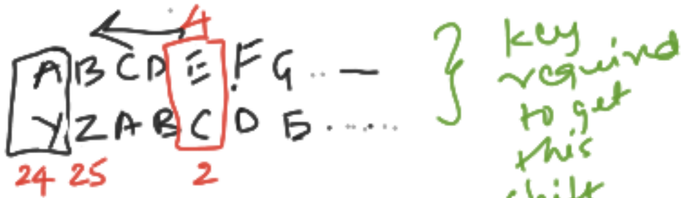



```
public int sum (int x, int y){
    return x & y;
}

public void countWL(FRAC, int[]
                        count){
```

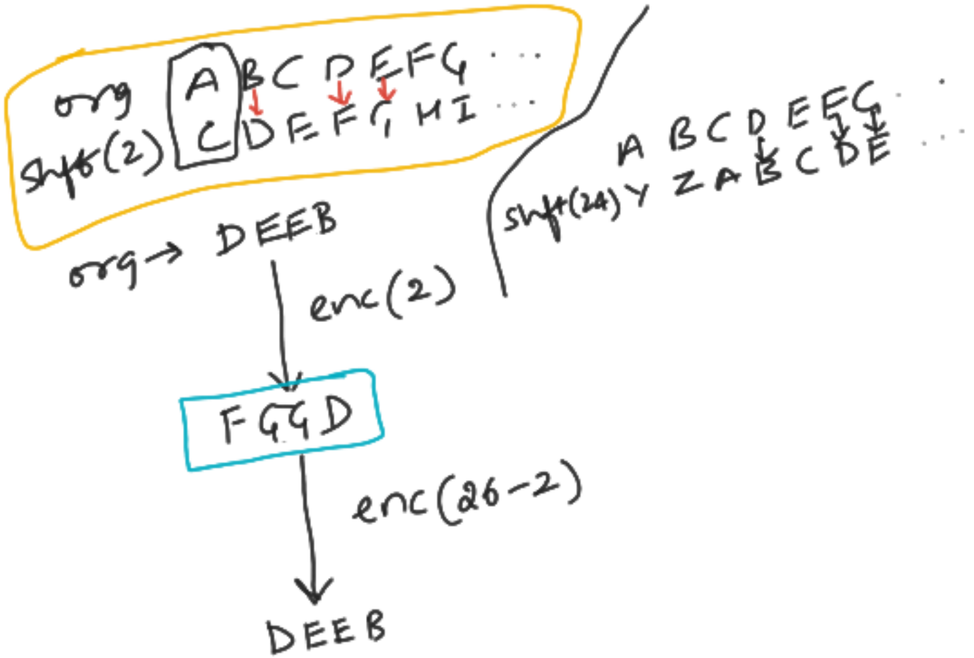


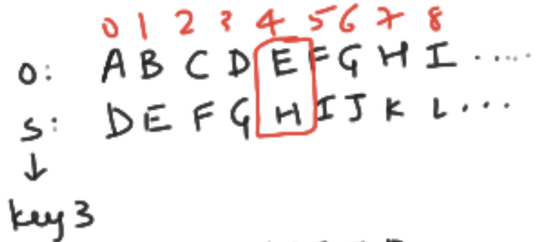
(2)

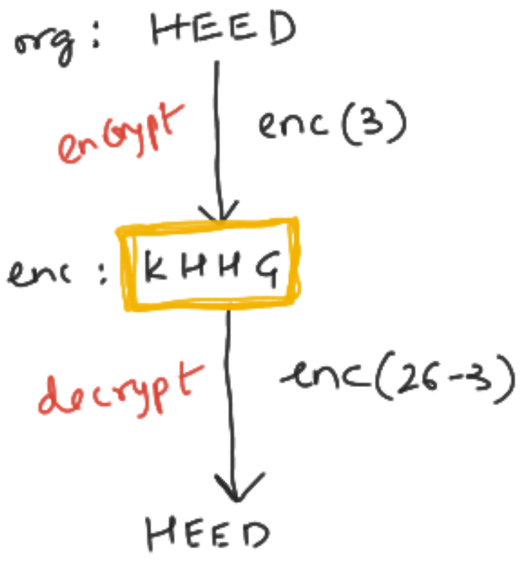


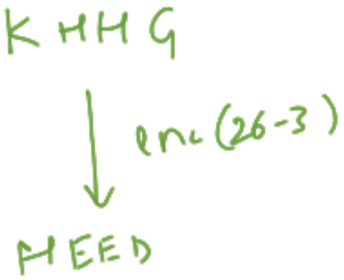
key required to get this shift

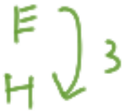
$26 - (4 - 2) = 24$

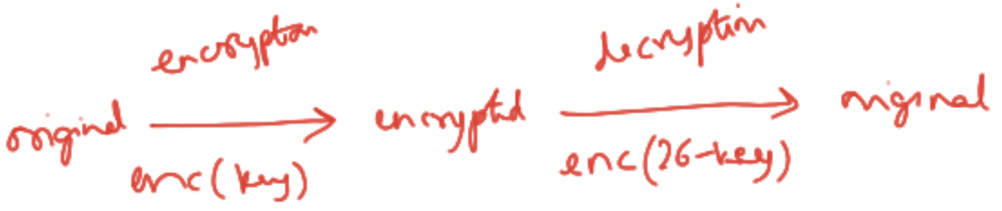



0 1 2 3 4 5 6 7 8
0: A B C D E F G H I .....
5: D E F G H I J K L...
key 3


org: HEED
encrypt
enc(3)
enc: KHHG
decrypt
enc(26-3)
HEED


KHHG
enc(26-3)
HEED


E
H
3


encryption
original
enc(key)
encrypted
decryption
enc(26-key)
original


```
String even = "";   SB even = new SB();
String odd = "";    SB odd = new SB();
for (i=0; i<s.len(); i++){
        char c = s.charAt(i),
        if (i%2 == 1) {
            odd += c;        odd.append(c);
        }
        else {
            even += c;       even.append(c);
        }
}
```

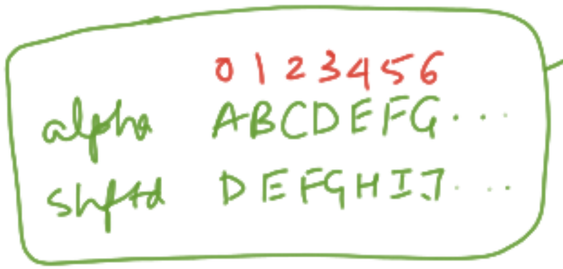
CaesarCipher cc

```
cc.encrypt(msg, key);
cc.decrypt(encMsg, 26-key)
```

```
CaesarCipher cc1 = new
                   CaesarCipher
                   (3);
```


```
cc1.encrypt(msg);

cc1.decrypt(encMsg);
```



based on key 3

BAD

↓

EDG

↓

BAD

```
class CaesarCipher {
    String alph;
    String shiftdAlph;
    int mainKey;

    public CaesarCipher(int key) {
        alph = " . ";
        shiftdAlph = . . . ;
        mainKey = key;
    }

    public String encrypt(String msg) {
        . . .
        . . .
    }

    public String decrypt(String enc) {
        CaesarCipher cc =
            new CaesarCipher(26 - mainKey);
        return cc.encrypt(enc);
    }
}
```

```
CaesarCipher cc =
        new CaesarCipher(3);
enc = cc.encrypt(msg);
cc.decrypt(enc);
```

alph ABCDE
shf DEFGH

CC2

alph ABCDE
shf XYZAB

```
class Math {
    public int comb(int n, int r){
     return (fact(n))/(fact(r) * fact(n-r)) ;

    }

    private int fact    (int n) {
        ...
    }
}
```

```
class TestMath {
    main() {
        // 10C3
        Math m = new Math();
        int res = m.comb(10,3);
    }
}
```

```
class BankAccount {
    private int bal;

    public void deposit(int x) {
        // some checks & validations
        bal = bal + x;
    }

    public int getBal() {
        return bal;
    }
}
```

```
BankAccount b = new
                BankAccount();

b.bal = 1000000;
b.deposit(-)
b.getBal();
```