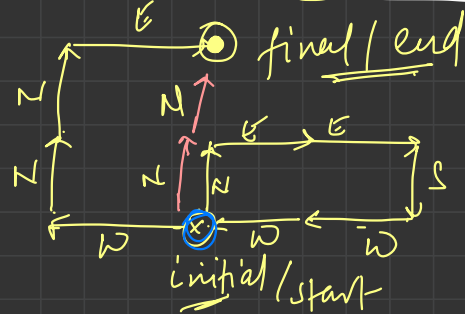
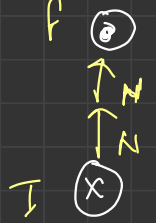
Lecture 8

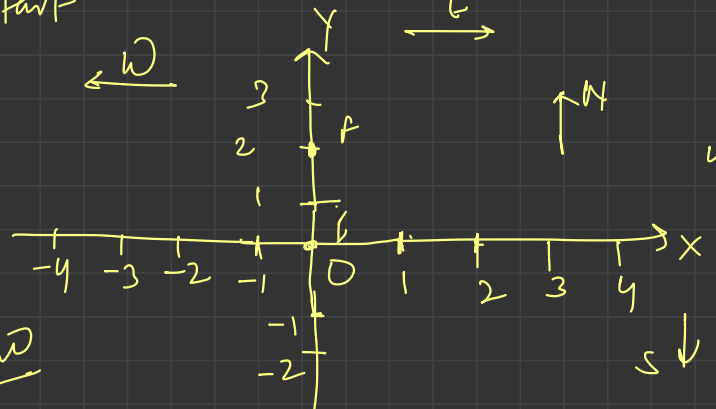
NESW

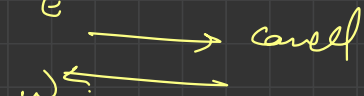
ID

NEESWWWNNE → NN (equivalent)









N → Y+1

S → Y−1

E → X+1

W → X−1

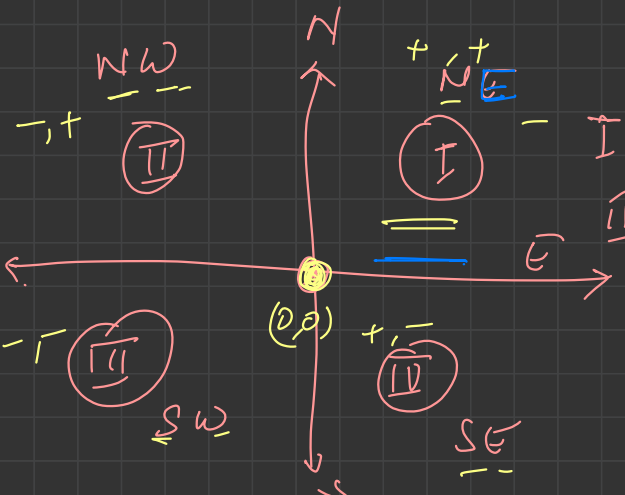
assume, we start at (0,0)

		X	Y
		0	0
Y+1	N	0	1
X+1	E	1	1
X+1	E	2	1
Y-1	S	2	0
X+1	W	1	0
X-1	W	0	0
X-1	W	-1	0
Y+1	N	-1	1
Y+1	N	-1	2
X+1	E	0	2

$\Rightarrow$ final position

$N \rightarrow Y+1$
$S \rightarrow Y-1$
$E \rightarrow X+1$
$W \rightarrow X-1$

(0,0) $\rightarrow$ (0,2)



I $\rightarrow$ N, E
II $\rightarrow$ N, W
III $\rightarrow$ S, W
IV $\rightarrow$ S, E

Final
(0,0) $\rightarrow$ no steps needed

only 2 of {N, E, S, W}

movement in x-dir : {E, W} → $x_f - x_i$

+ve → go east

-ve → go west

$|x_f - x_i|$ times

y-dir : {N, S} → $y_f - y_i$

+ve → go north

-ve → go south

$|y_f - y_i|$ times

I (0, 0) F (0, 2)

E,W 2-0 → 2 ⇒ 2 steps north ⇒ NN

F N,S

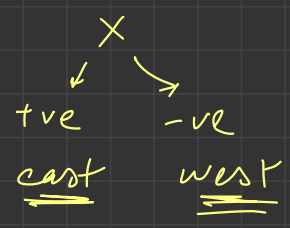
I (0,0) → (5,4) → 5E 4N

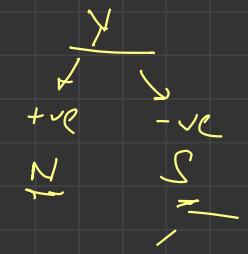
(-3,2) → 3W 2N

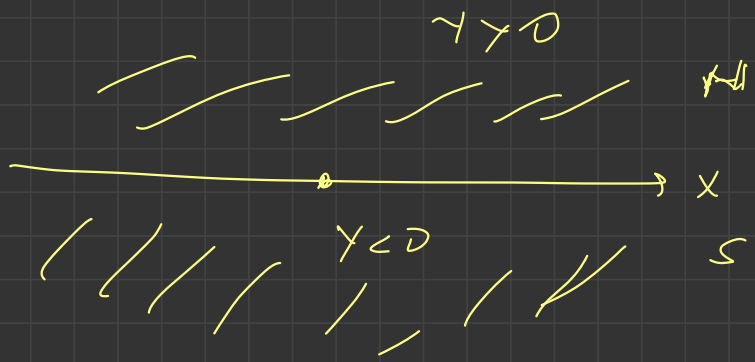
II (2,-1) → 2E 1S

(-1,4) → 1W 4N

(-6,-1) → 6W 1S






y>0
N
x
y<0
S

Q.1. Count number of words, characters, lines when input ends with $.

today is friday,
and it is also
Jan, 13, 2023$

10 words?
Char ?
3 lines ?

Q.2. Count number of digits, whitespace, uppercase, lowercase and special characters (except $) ending with $.

Array → declaration

```
int arr[n];
char carr[n];
bool barr[n];
double - - - -
```

Array initialization

```
int arr[5] = {1,2,3,4,5};
```

arr[0] → 1
arr[1] → 2
arr[2] → 3
arr[3] → 4
arr[4] → 5

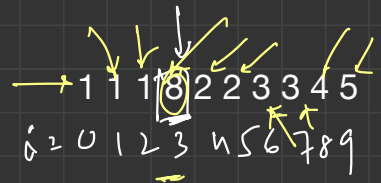
initialize ?!

```
int arr[ ] = {2,4,9};
```

size = 3

arr[0] = 2
arr[1] = 4
arr[2] = 9

input → storing in each index.



8 → unique

i = 3

→ 8
→ 4
→ 5

① pick an element from the array. ⇒ arr[i]

② check the entire whether, same value is present at a different index.

arr[j] == arr[i] and i != j

index = i = 6

arr[i] = arr[6] = 3 ✗ not unique

1 1 1 & 2 2 ~~8~~ 3 → (7)

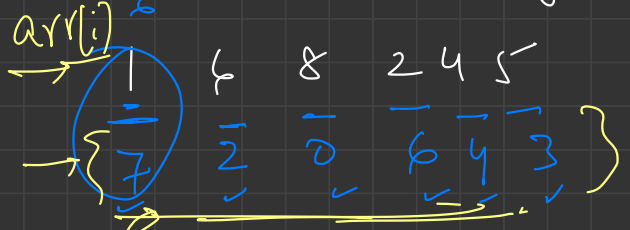
6

Assumption: All the values in the array are unique.

```
for(i=0; i<n; i++) {
    arr[i] ← picked element in (1)
    bool isUnique = true;
    for(j=0; j<n ; j++) {
        if( arr[i] == arr[j] && i != j) {
            isUnique = false;
        }
    }
    if(isUnique == true) {
```

```
        cout<< arr[i] << ' ';
    }
}
```

Assumption: Any value can be the part of the pair.



→ X = 8

X - arr[i] → ① pick element

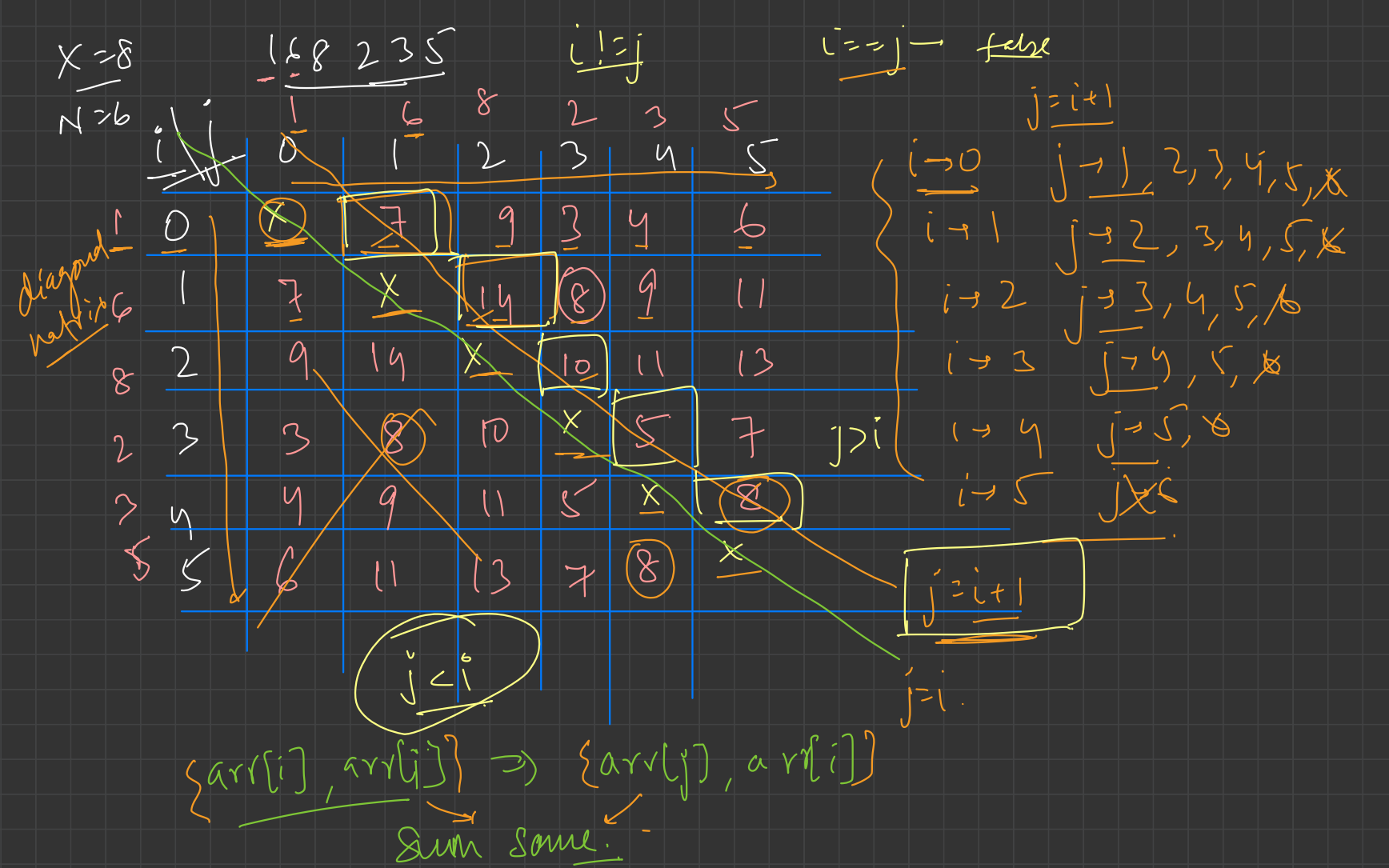
② check whether (X - arr[i]) exists at $j \neq i$.

true → pair

Solved!

```
for(i=0; i<n; i++) {
    for(j=0; j<n; j++) {
        if( i!=j && arr[i]+arr[j]==x ){
            cout<<arr[i]<<' '<<arr[j]<<endl;
        }
    }
}
```

arr[j] == x - arr[i]


X=8
N=6
1 6 8 2 3 5
i!=j
i==j → false
j=i+1
i→0 j→1,2,3,4,5,✗
i→1 j→2,3,4,5,✗
i→2 j→3,4,5,✗
i→3 j→4,5,✗
i→4 j→5,✗
i→5 j✗✗
j>i
diagonal half is 6
j<i
j=i+1
j=1.
{arr[i], arr[j]} ⇒ {arr[j], arr[i]}
Sum same.