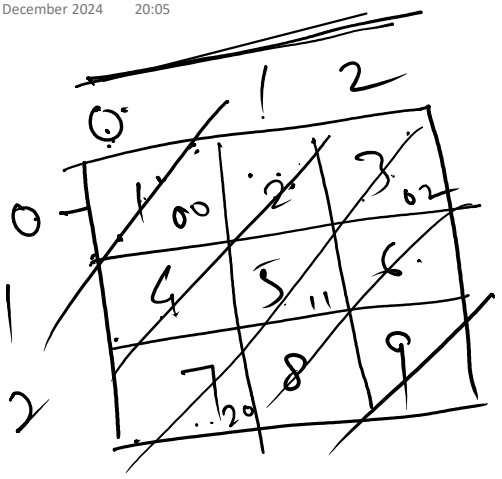


1, 2, 4, 3, 5, 7, 6, 8, 9

0 to 2.

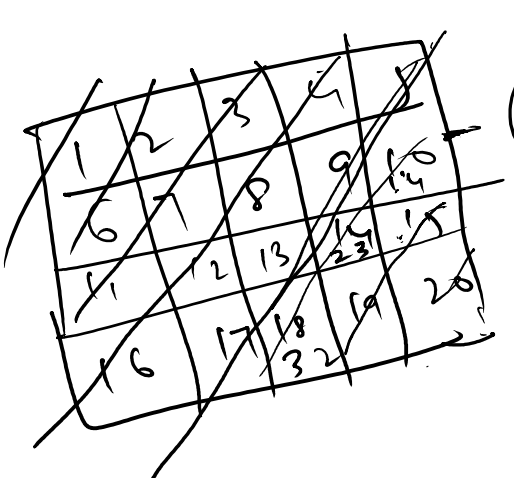
0 row start → 0 to col 00

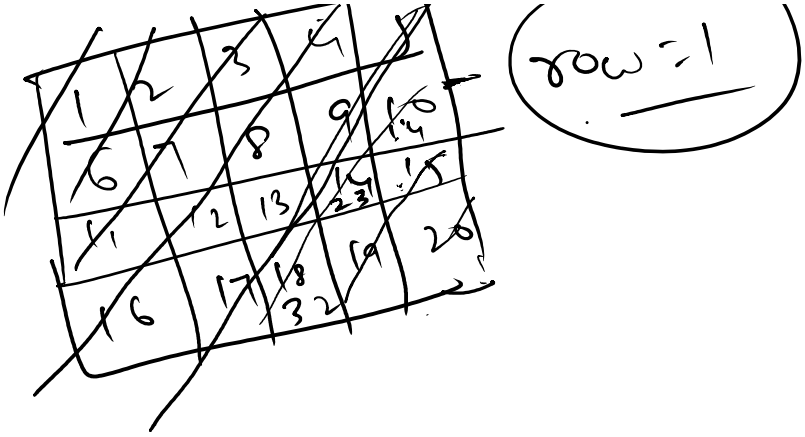
1. row → 0 to col 01 10

row → 1

2 . 02 20

```
for( int j=0; j<col; j++) {
int left=0, int colvar=j
While(left<=colvar){
S.O.P(arr[left][colvar]
left++, colvar--;
```




row = 1

String:- It is collection of characters

We represent string by double qoutes.

'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i',

L> These are characters

"JayKumar" → 'J', 'a', 'y', 'K', 'u', 'm', 'a', 'r'

Properties:-(i). String is an object

(ii) Strings are immutable that means it can not be modified.

```
String name = "Izhar";
    name = "Izhar" + "haider";
```

This won't give any syntax error, but internally it creates a new object after modification.

To compare two strings:-

(i) Using equals() method.

```
String name = "Jay";
String name2 = "Jay";
```

name.equals(name2) → this will return true

Reason:- equals method only compares the value stored in string object.

(ii) using (==)

```
String name1 = "Jay";
String name2 = "Jay";
```

name1 == name2 → it will return false

It compares both → value as well as object

name1 == "Jay" → true

Change string to char

Change string to Char

```
String name = "Jay";
name.charAt(0) → This will return 'J' as a character.
```

(i.) charAt(index) → It will return the character present at the index

(ii.) Length of String → length()
This method will give length of string

Print all character of string

```
String name = "Jay";
for(int i=0; i<name.length(); i++){
  S.O.Pln(name.charAt(i));
}
```

(iii.) toCharArray() → Using this method we can convert string to char array

```
Char ch[] = name.toCharArray();
```

(iv.) toUpperCase() → convert all characters of string to upper case

```
String name = "Jay";
S.O.Pln(name.toUpperCase());
```

Output → JAY

(v-) toLowerCase() → convert all characters of sting to lower case

```
String name = "Jay";
S.O.P(name.toLowerCase());
```

Output :- jay

Question :-

1. Print all characters of string
2. Check if two strings are equal or not

3. Print index of vowel characters
4. Count words in a sentence.

trim() :- It removes starting and ending spaces.

```
String name:" Jay ";
  S.o.pln(name.trim())
```

Output → Jay

split() :- It breaks the string into array of strings, parameter of this function is delimiter.

```
String intro="My name is Jay Kumar";
String str[] = intro.split(" ");
//str[] = ["My", "name", "is", "Jay", "Kumar"];
for(int i=0; i<str.length; i++){
   s.o.pln(str[i]);
}
```

```
String sentence: "All/is/well";
String str[] = sentence.split("/");
```

↳ "All", "is", "well"

split() → Time Complexity → O(n)
↳ length of string

When we use split
split(" ") → It means single space.
split("\\s+") → It will divide the

✓ split("\\s+") → It will divide the string irrespective of no. of spaces.