Note Title 26-07-2018

COL 100
July 26, 2018

Last Class:-

1. Computers? [0--1 0--1]
2. Programming Language [C++]
3. Algorithms:-

Efficient

29 10138769

$\sqrt{29}$

Check whether

For i = 2 to 28 { 1 ✗ goes from 2 ✗ 28

if (i divides 29)

→ then not prime();

→ prime();

}

Print all prime from 1 ✗ 100:-

Problem :=

Algorithm :-

① An algorithm is a step by step procedure to solve a problem.

② → Step by step refinement of the problem at hand;

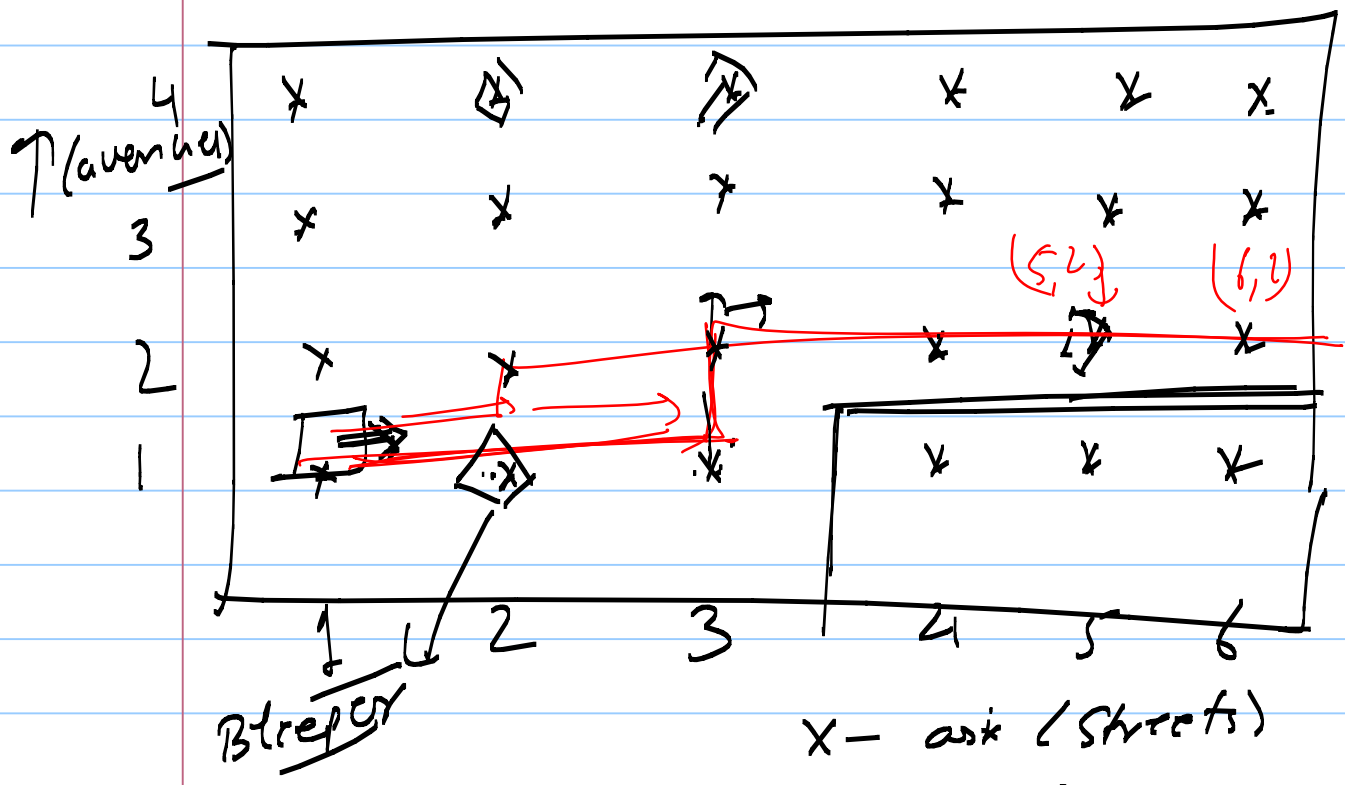
---

pseudocode

```
Print N Primes { n ← 100; 1000;
    Go from
    For i = 1 to n
        if not prime(n) then
            print n;
}
```

Computer Science → Correct
→ Efficient
→ Fast
→ memory

Robot :- Karel




```
move();
turnLeft();
```

```
pickBreeper();
putBeeper();
```

→ function

moveAndGetBeeper() { → start

move(4); argument

pickBeeper();

move();

} → end

moveSteps(n

→ with comp back

~~move_to~~

~~move Far~~

moveBeeperFar() {

move();

pickBeeper();

move();

turnLeft();

move();

turnLeft();

turnLeft();

move();

} turnright();

turnRight() {

turnLeft();

turnLeft();

turnLeft();

}

move();

putBeeper();

move();

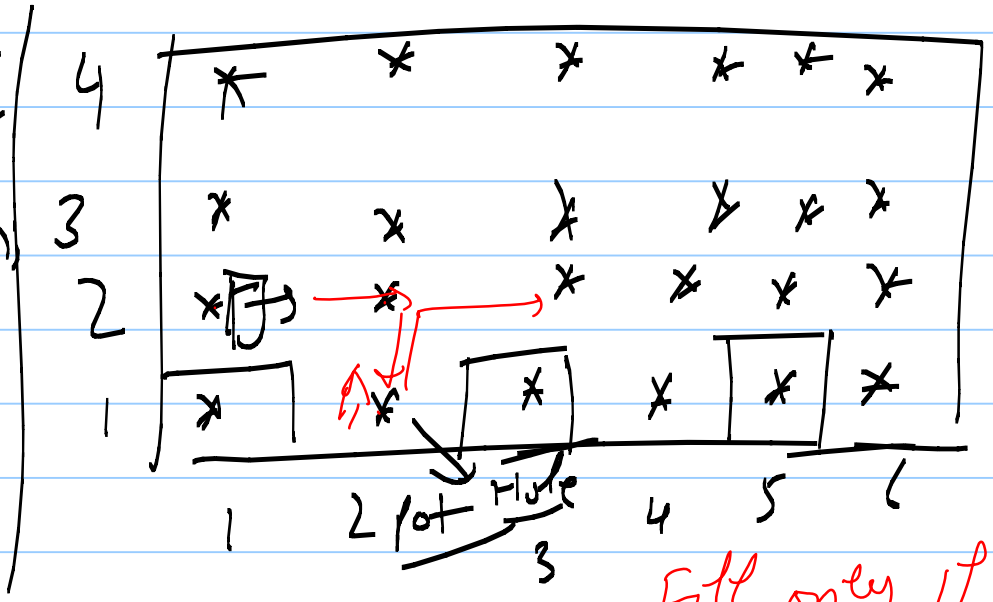
turnLeft();

turnLeft();

} turnAround();

```
moveAndFill(){
  move();
  fillPotHole();
  move();
}
```

3




```
moveAndFill() {
  move();
  turnRight();
  move();
  putBeeper();
  turnAround();
  move();
  turnRight();
  move();
}
```

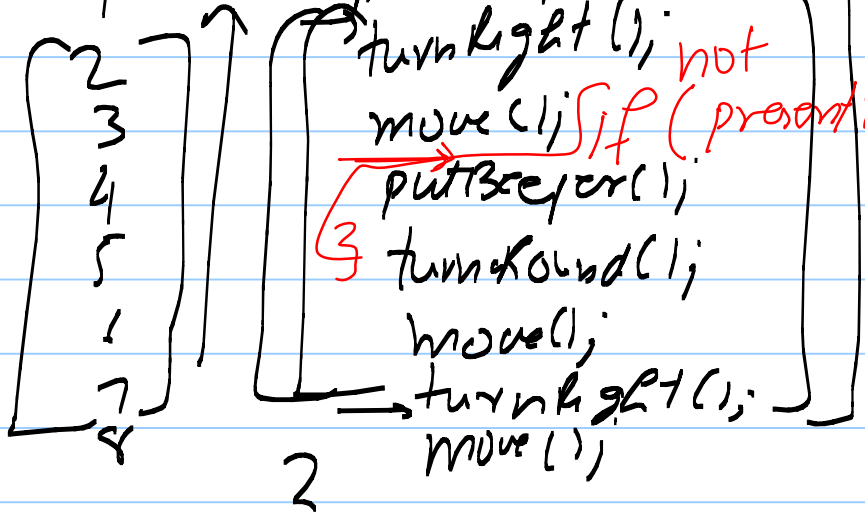
Fill only if no beeper is present

not

```
if (presentBeeper()) {
}
```

conditional

```
fillPotHole();
```



moveAllPotholes() {

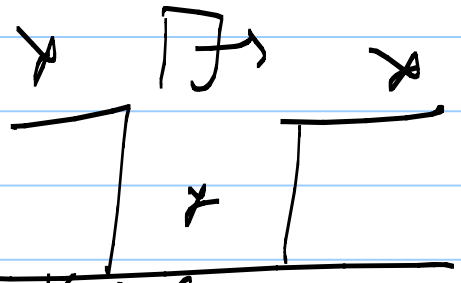
```
move();
fillPothole();
move();
move();
fillPothole();
move();
move();
fillPothole();
}
```

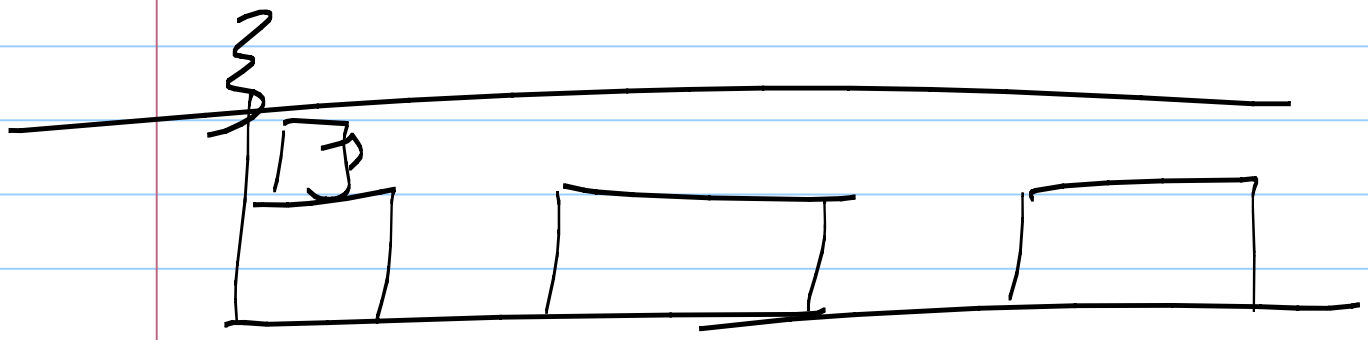
Conditionals:-

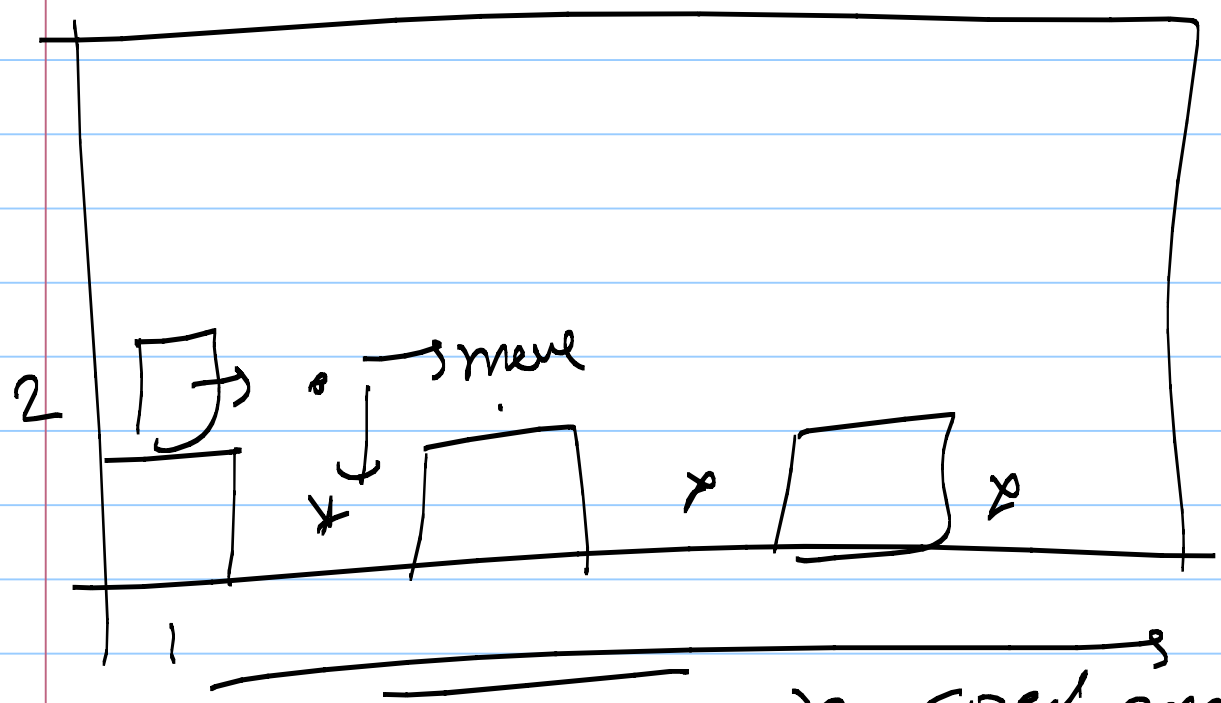
frontIsClear();
leftIsClear();
rightIsClear();
beeperPresent();
facingNorth();
facingEast();
facingWest();
facingSouth();

```
fillPothole() {
    turnRight();
    move();
    if (notBeeperPresent()) {
        putBeeper();
    }
    turnAround();
    move();
    TurnRight();
}
```

conditional







For i = 1 to n → n sized grid
→ Loop

```
{
    move();
    fillSpot();
}   move();
```