Lec4

Saturday, 5 June 2021 7:09 PM

```
import java.util.*;
public class Inverse {

public static void main(String[] arg
// TODO Auto-generated method stub
int n = 32145;
int idx = 1; ✓
int ans = 0;
while (n > 0) {
int rem = n % 10;
ans += (int) (idx * Math.pow(10, rem
n/=10; // n=n/10;
idx++;
}
System.out.println(ans);

}

}
```

double

log

32148 % 10

82143 ✓

10^n

10000

3

```
s) {
```

3 2 1 5

2 4 3

```
- 1));
```

5 4 3 2 1

3 2 1 4 5

n =

3 →

2 1 4 →

1 2 5 4 3

3 x 1

5-1

4-1

= 32145

-1 2-1

32145 % 10

32145/10 = $\boxed{3214}$

Pattern

N = 5

Row →

```
                    5
X X X X X
X X X X X
X X X X X
X X X X X
X X X X X
```

$\times 10$

$1 \times 10 + 2 \times 10 + 3 \times 10$

$S \times 10^{3-1}$

10-12

> NST

↓

Number of steps

CST

↓

CST → NST

① → ~~7H → 8H5~~ ✓

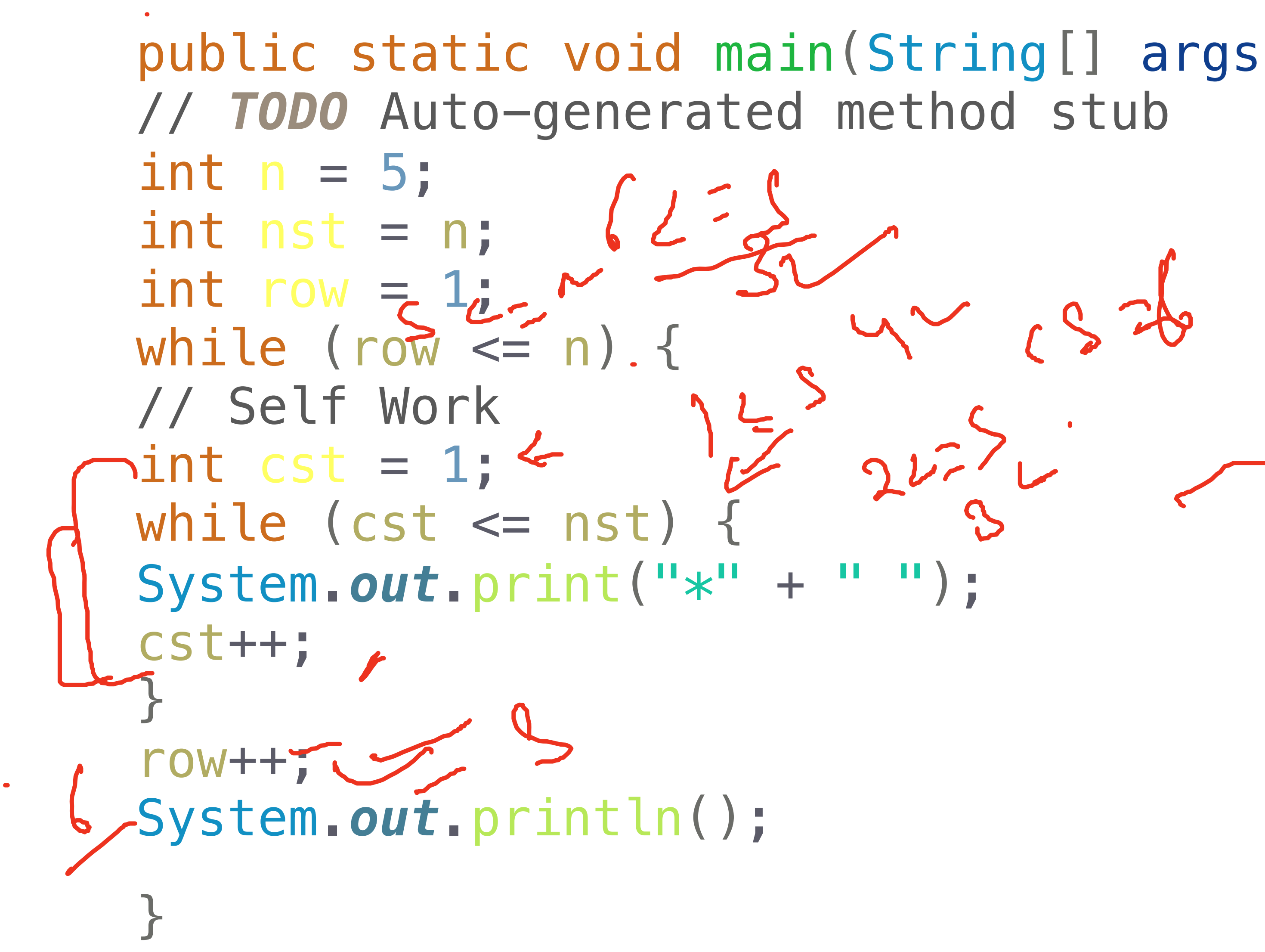
$+ 4 \times 10^{-}$

$-$

nst →

x x x

3



```
public static void main(String[] args
// TODO Auto-generated method stub
int n = 5;
int nst = n;
int row = 1;
while (row <= n) {
// Self Work
int cst = 1;
while (cst <= nst) {
System.out.print("*" + " ");
cst++;
}
row++;
System.out.println();

}
```

(i) ✓

1st decide

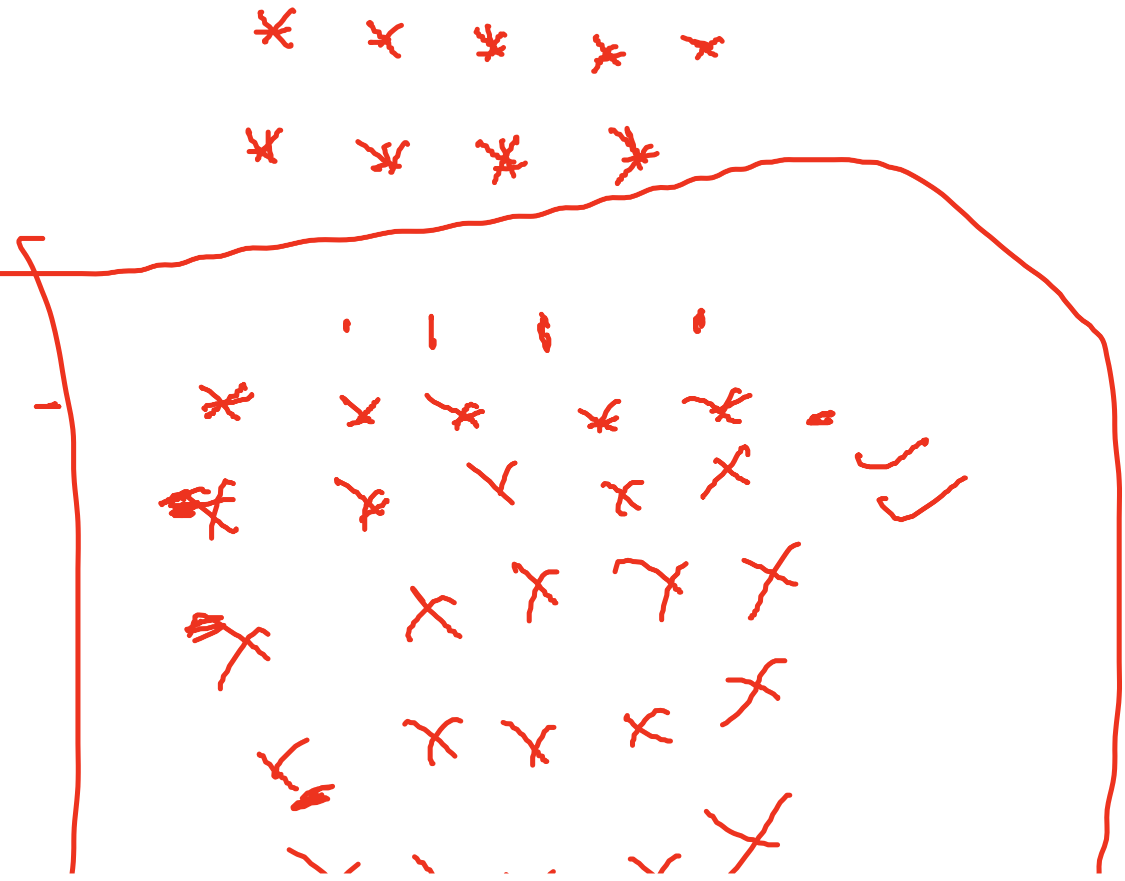
(ii) → Self work

(iii) Prep → for most

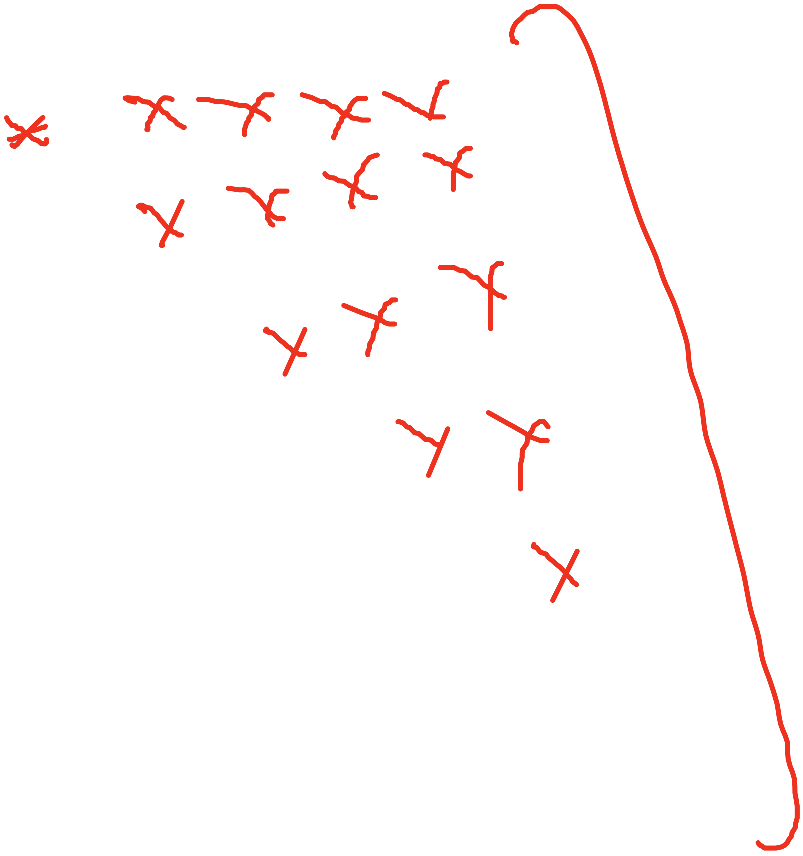
||x||| + || 4

) {

Row=2



```
}
```



```
while (row <= 2 * n -
int cst = 1;
while (cst <= nst) {
System.out.print("* "
cst++;
}
}
```

Row++ ✓

nst ✓

x x x x

Nst = n

nsp = 0

Row = 1

NST = 1

3

nST → 3

1 2 | 2

1 2

- 1) {

Row

x

x x

x x x

x x x x →

');

Row

nST

Row -> nst=

Row=6

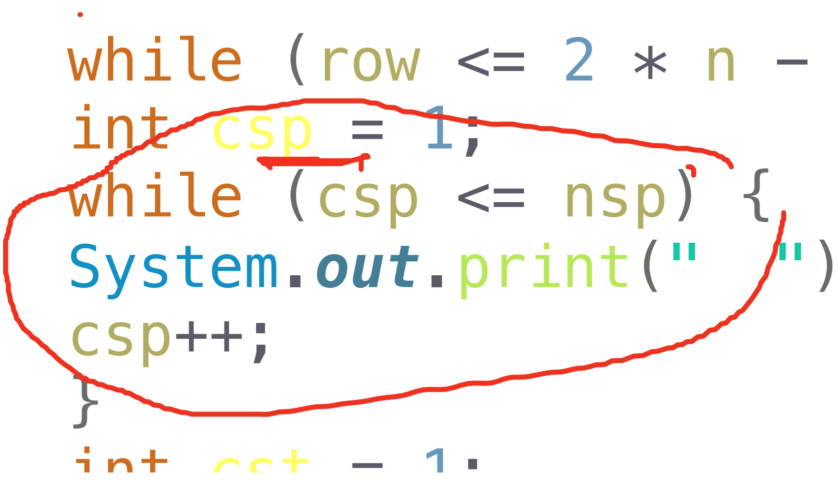
if (Ro

n

}

Nsp=n-1

nst



```
while (row <= 2 * n -
int csp = 1;
while (csp <= nsp) {
System.out.print(" ")
csp++;
}
int cst = 1;
```

S

x x x x x

x x x x

→ Row

nst

2n)(

s++

else {

nst--

}

4

Row = 1 ≤ 3

1) {

- - - - x

- - - x x

;

- - x x x

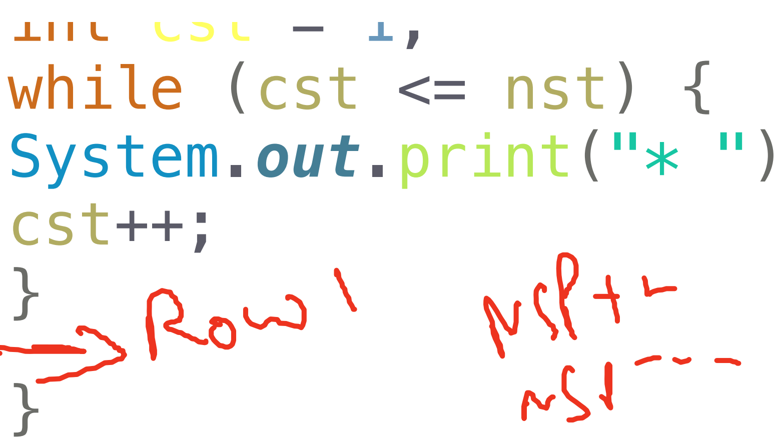
~~nst = 3~~

Row

- x x x x

nst = 4

—



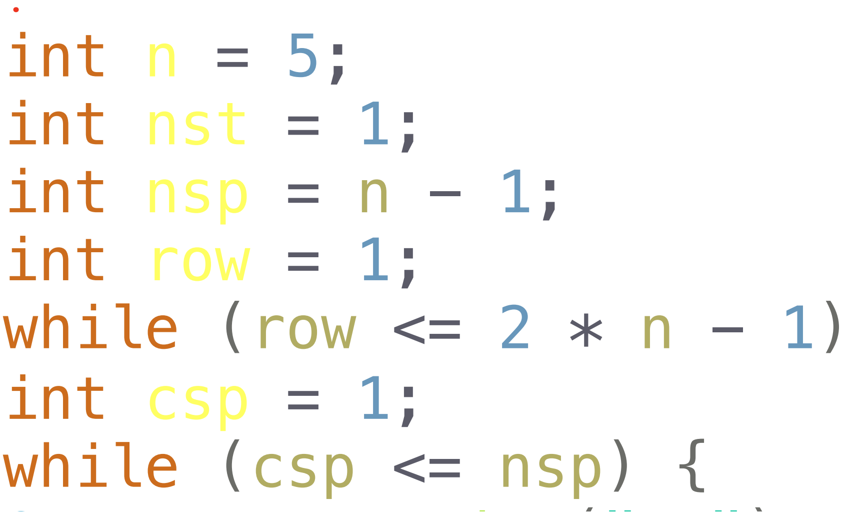
```
int cst = 1;
while (cst <= nst) {
System.out.print("* ")
cst++;
}

}
```

Row 1

nsp++
nst--

if (Row<=n) {
nst++;
nsp--;
}



```
int n = 5;
int nst = 1;
int nsp = n - 1;
int row = 1;
while (row <= 2 * n - 1)
int csp = 1;
while (csp <= nsp) {
```

;

Row { Nst--
N++1

{

```
System.out.print("  ");
csp++;
}
int cst = 1;
while (cst <= nst) {
System.out.print("* ");
cst++;
}
if(row<n) {
nsp--;
nst++;
}
else {
nsp++;
nst--;
}
row++;
System.out.println();

}

}

}
```