

→ functions →

→ Diagonal printing →

→

factorial
9!
5!

public static int factorial (int n)

```
int ans1 = factorial(n); → 9;
int ans2 = factorial(r); → 5;
int ans3 = factorial(n-r); → 4!
```

1st ans = $\frac{1}{(ans + 1)}$

Stack

```
public static int factorial(int n){
```

```
    int ans=1;
```

```
    for(int i=1;i<=n;i++){
        ans*=i;
    }
```

```
    return ans;
```

```
}
```

```
public static void main(String[] args) {
```

```
    /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
```

```
    int n=9;
```

```
    int r=5;
```

```
    int ans1=factorial(n);
```

```
    int ans2=factorial(r);
```

```
    int ans3=factorial(n-r);
```

```
    int ans=(ans1)/(ans2*ans3);
```

```
    System.out.println(ans);
```

Memory Mapping

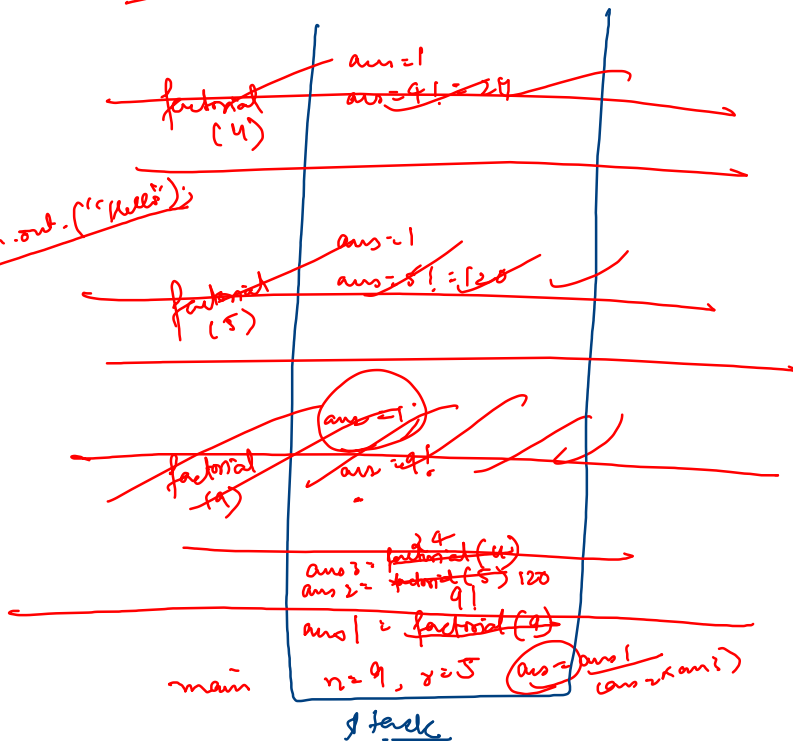
fn

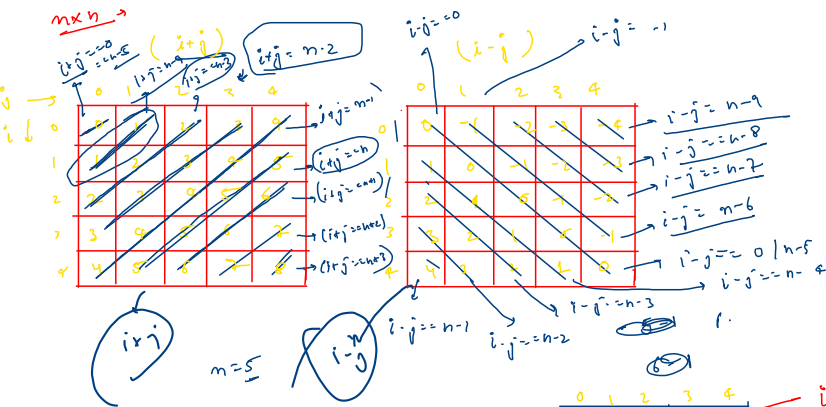
System.out.println("Hello");

void return

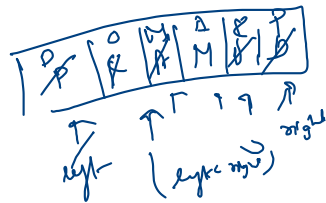
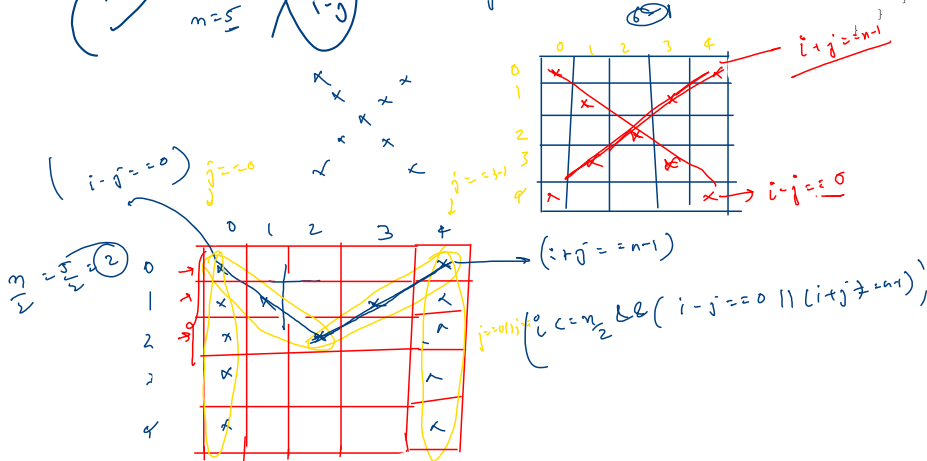
public static boolean fn (int a) {
 return

return





```
public class Solution {
    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to
        int n=5;
        for(int i=0;i<5;i++){
            for(int j=0;j<5;j++){
                if(j==0 || j==n-1 || (i<=n/2 && (i+j==n-1 || i-j==0))){
                    System.out.print("X");
                }else{
                    System.out.print(" ");
                }
            }
            System.out.println();
        }
    }
}
```



while (left < right)

