Fun calling itself / Fun calling

A(){ A()✓ }

A(){ B()✓ }

PMI:

$\sum_{i=1}^{i=N} i = 1+2+3+\dots+N = \frac{N(N+1)}{2}$

$\Rightarrow 1 = \frac{1 \times (1+1)}{2}$

$\Rightarrow 1+2 = \frac{2 \times (2+1)}{2}$

$\sum_{i=1}^{i=K} i = 1+2+3+\dots+K = \frac{K(K+1)}{2}$ ✓

$\sum_{i=1}^{i=K+1} i = 1+2+3+4+\dots+K + K+1 = \frac{(K+1)(K+2)}{2}$

$\frac{K(K+1)}{2} + (K+1) = \frac{(K+1)(K+2)}{2}$

(i) smaller-input ✓
(ii) N=K ✓ correct ✓

$(K+1)(\frac{K}{2}+1) = \frac{(K+1)(K+2)}{2}$

$\frac{(K+1)(K+2)}{2}$

(i) smaller input
(ii) S.P ✓
(iii) self work

Recursion पहला प्यार है !!

(1) $N! = (N-1)! \times N$

$5! = 4! \times 5$ (24 → 120)

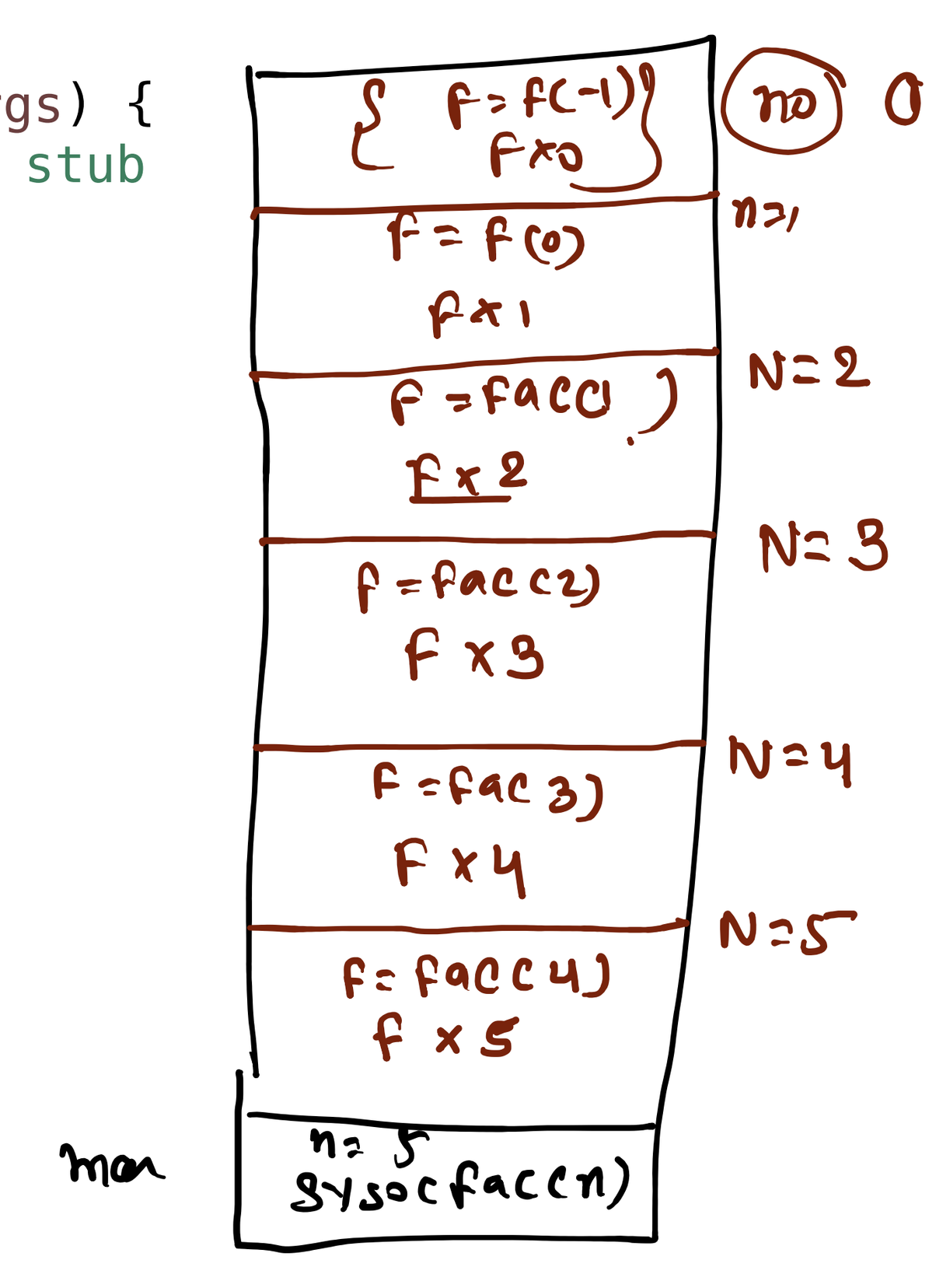
$3! \times 4$ (6)

$2! \times 3$ (2)

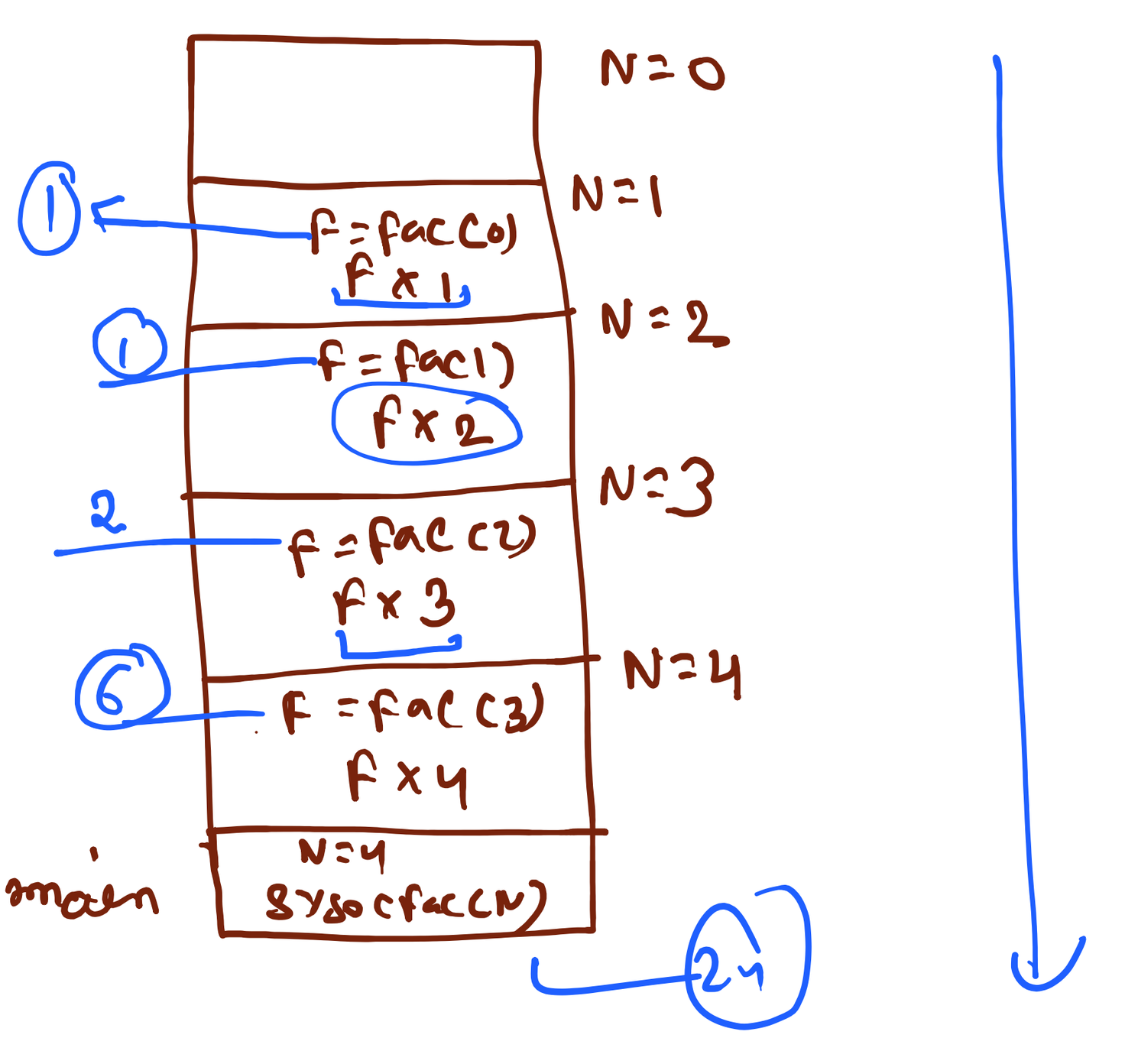
$1 — 1! \times 2$

$1 \leftarrow 0! \times 1$

```java
public static void main(String[] args) {
	// TODO Auto-generated method stub
	int n = 5;
	System.out.println(fac(n));
}
public static int fac(int n) {
	int f = fac(n - 1);// SP
	return f * n;
}
```

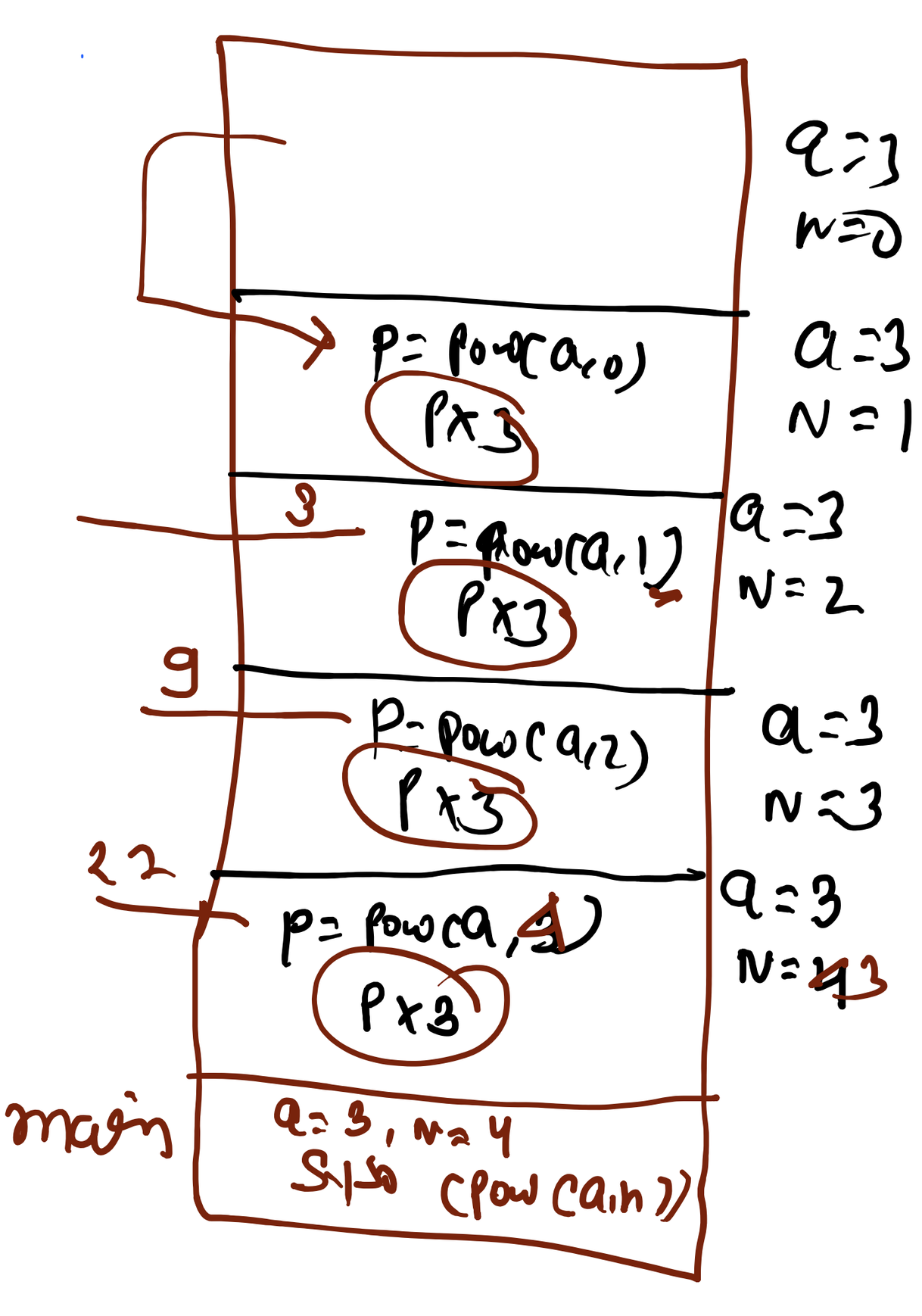


```java
public static int fac(int n) {
	// Base Case
	if (n == 0) {
		return 1;
	}
	int f = fac(n - 1);// SP
	return f * n;
}
```

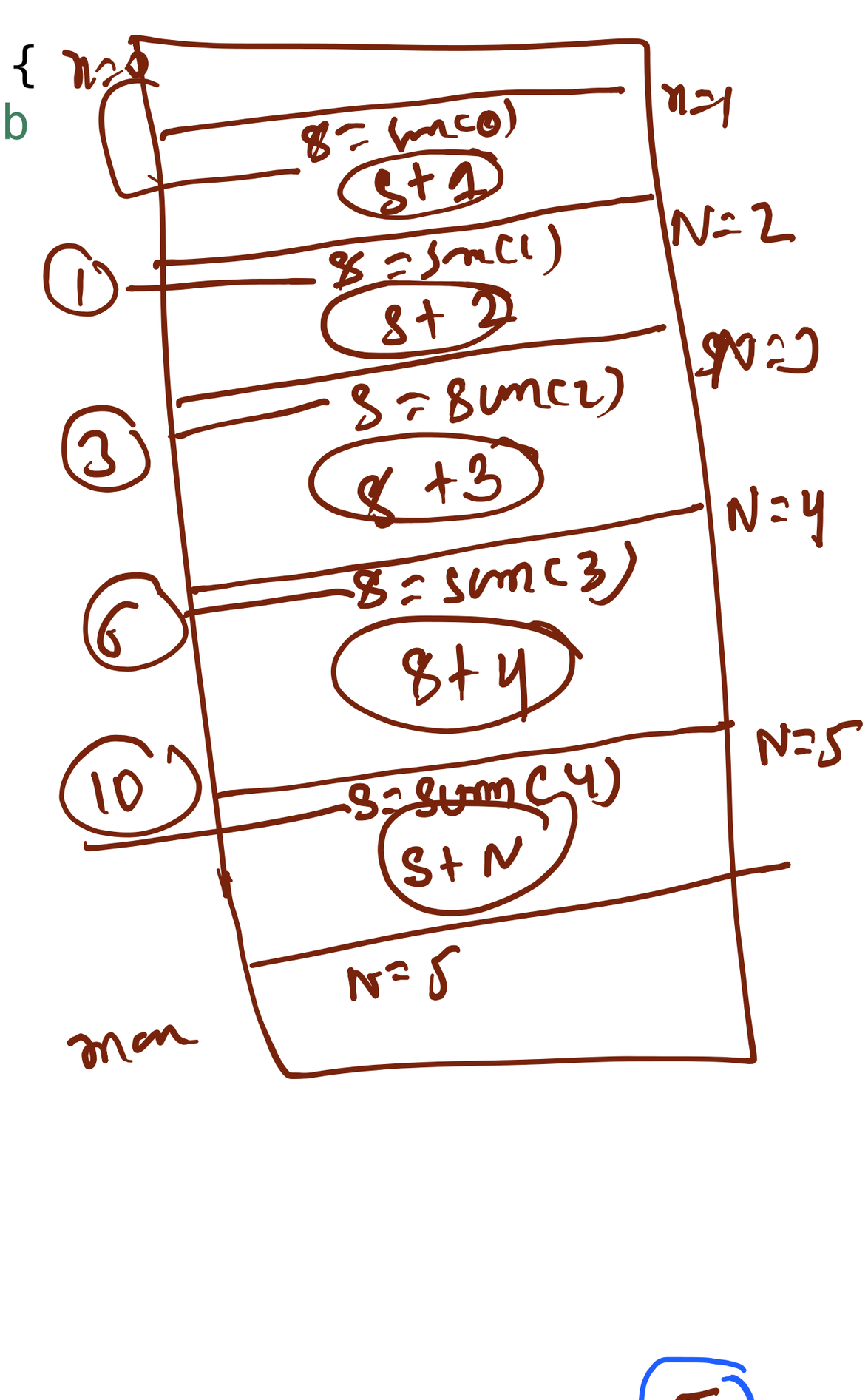


a^n

```java
public static void main(String[] args) {
	// TODO Auto-generated method stub
	int a = 3;
	int n = 4;
	System.out.println(pow(a, n));
}
public static int pow(int a, int n) {
	if (n == 0) {
		return 1;
	}
	int p = pow(a, n - 1);// sp
	return p * a;
}
```



```java
public static void main(String[] args) {
	// TODO Auto-generated method stub
	int n=5;
}
public static int Sum(int n) {
	if(n==0) {
		return 0;
	}
	int s=Sum(n-1);
	return s+n;
}
```



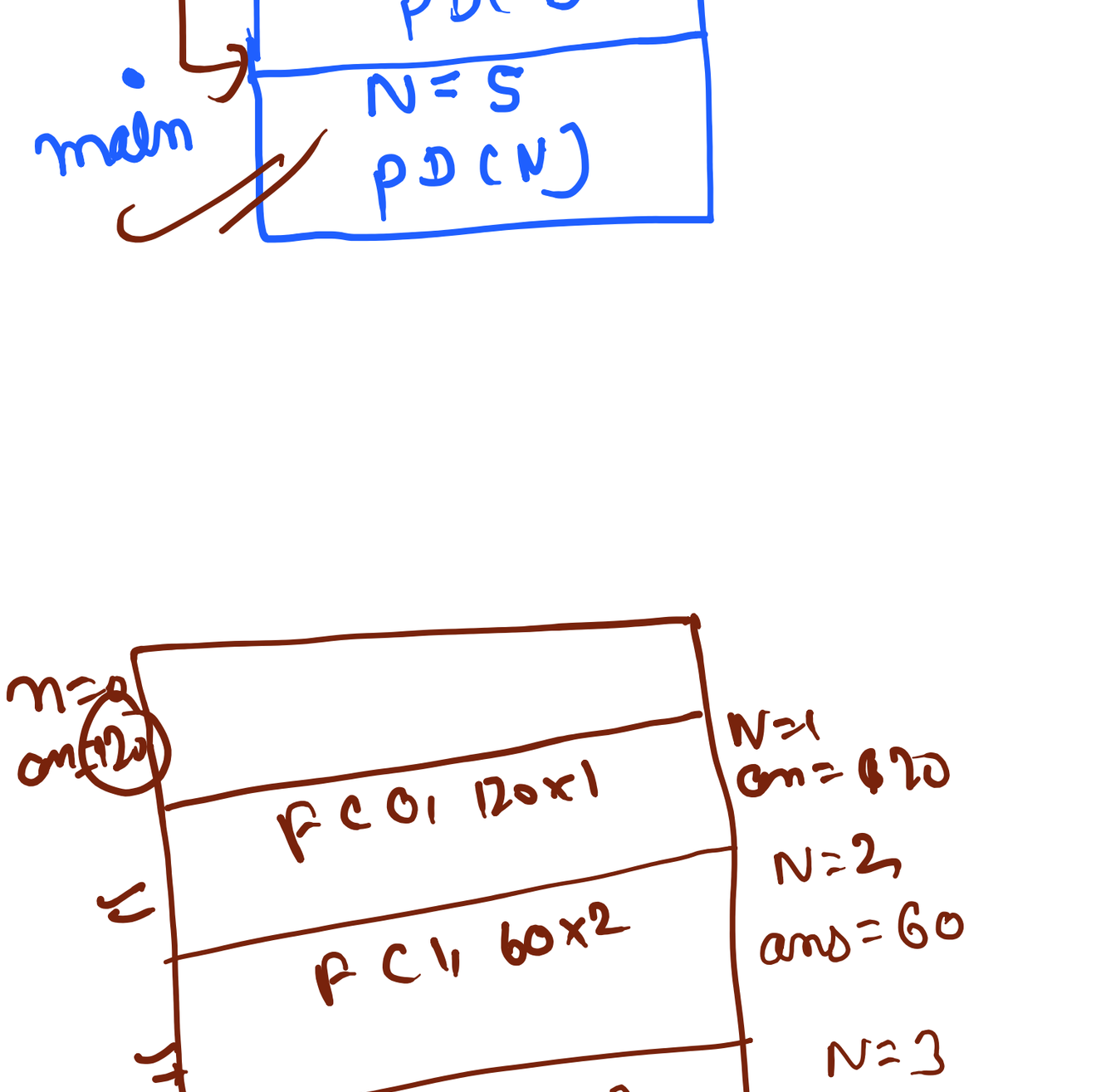
N = 5 4 3 2 1

N=5

5
4
3
2
1

PD(N-1)

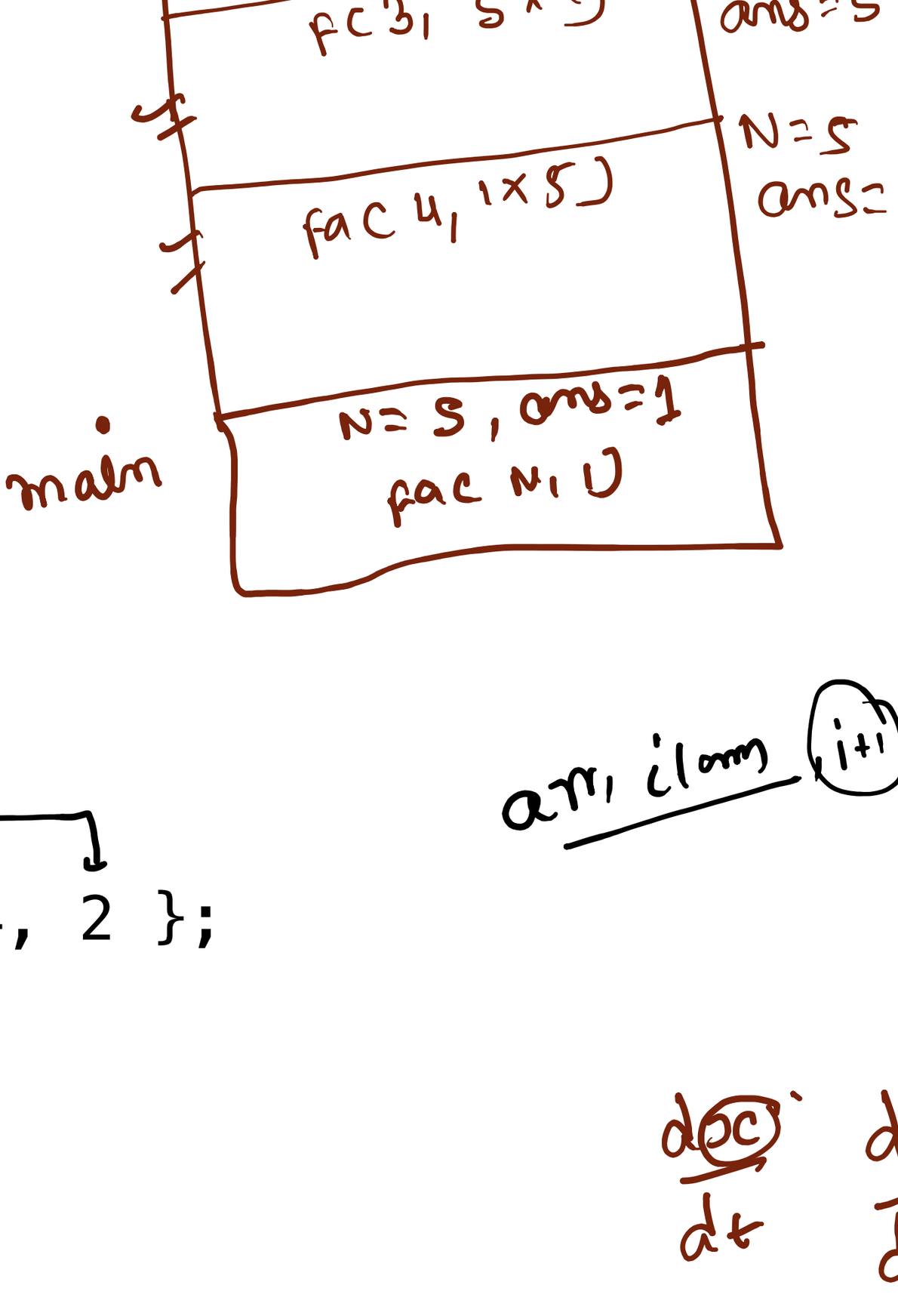
```java
public static void PD(int n) {
	// TODO Auto-generated method stub
	if (n == 0) {
		return;
	}
	System.out.println(n);
	PD(n - 1);
}
```



5
4
3
2
1

1
2
3
4

```java
public static int fac(int n,int ans) {
	// Base Case
	if (n == 0) {
		return 1;
	}
	return fac(n-1, ans*n);
}
```



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

item

arr, item (i+1)

$\frac{d(c)}{dt}$ $\frac{dc}{dt} \Rightarrow 0$

```java
public static void main(String[] args) {
	// TODO Auto-generated method stub
	int[] arr = { 3, 4, 2, 4, 2 };
	int item = 4;
}
public static int Index(int [] arr,int item,int i) {

}
```

0 1 2 3 4

```java
public static int Index(int[] arr, int item, int i) {
	if (i == arr.length) {
		return -1;
	}
	if (arr[i] == item) {
		return i;
	}
	return Index(arr, item, i + 1);
}
```

-ve

2K → | 3 | 2 | 4 | 3 |
0 1