

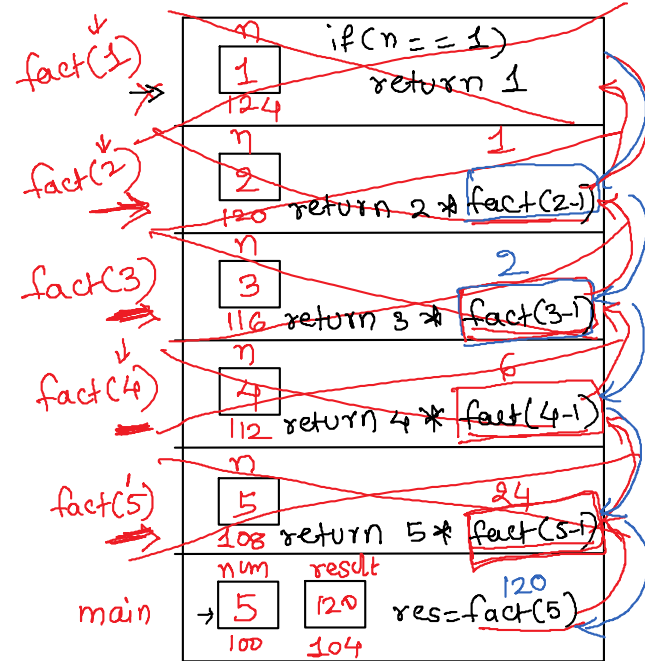
```

✓ int factorial(int n)
{
    ✓ if(n==1)
        return 1;
    return n * factorial(n-1);
}

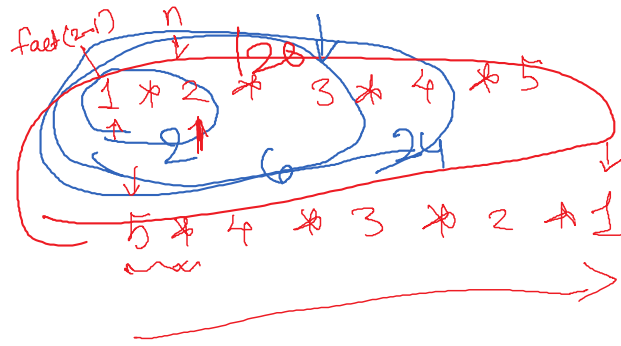
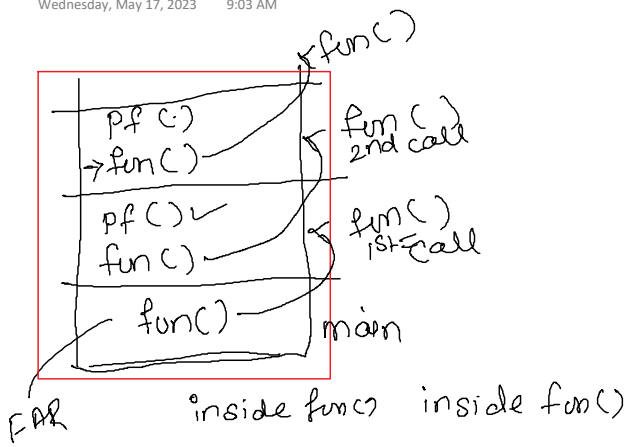
int main()
{
    ✓ int num, result;
    ✓ printf("Specify number \n");
    ✓ scanf("%d", &num);
    result = factorial(num);
    printf("Factorial of given number is %d\n", result);
}

```

120



<div></div>
<div></div>
<div></div>
<div></div>
<div></div>
<div><div></div><div></div></div>



$$\begin{aligned}
 2 * \text{fact}(2-1) &= \text{fact}(1) = 2 \\
 3 * \text{fact}(3-1) / \text{fact}(2) &= 6 \\
 4 * \text{fact}(4-1) / \text{fact}(3) &= 24 \\
 5 * \text{fact}(5-1) / \text{fact}(4) &= 120
 \end{aligned}$$

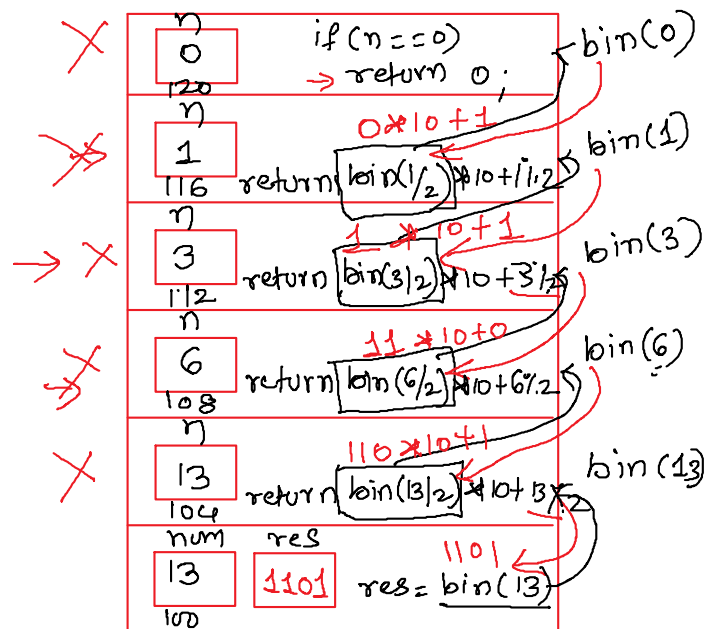
```

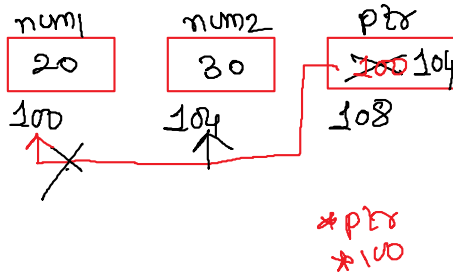
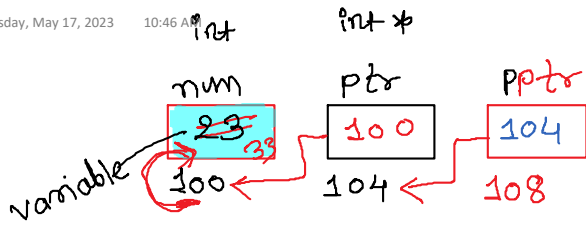
#include<stdio.h>
int binary(int n)
{
    if(n==0)
        return 0;
    return binary(n/2) * 10 + (n%2);
}

int main()
{
    int num,result;
    printf("Specify number \n");
    scanf("%d",&num);
    result = binary(num);
    printf("Binary of given number is %d \n",result);
}

```

m





$pptr = 104$ — add of pointer to int

$\& pptr = 108$

$* pptr$

$*(104) = 100$ — after 1 indirection
level add of int

$** pptr$

$(*) (104)$

$(*) (100)$

23

```
int main()
{
    int num = 23;
    printf("num = %d \n", num);
    fun(&num);
}

void fun(int *ptr)
{
    ✓ *ptr = *ptr + 10;
}
```

$\star(100)$
 $23 + 10$
 $\star ptr = 33$

