

```

void printNum(int n) // recurs:
{
    //base case
    if (n==0) //n=4
    {
        return;
    }
    printNum(n-1); // 4 - 1 = 3
    printf("%d ", n);
}

```

```

void printNum(int n) // recursive funct
{
    //base case
    if (n==0) //n=3
    {
        return;
    }
    printNum(n-1); // n - 1 = 2 //callin
    printf("%d ", n);
}

```

```

void printNum(int n) // recursive 1
{
    //base case
    if (n==0) //n=2
    {
        return;
    }
    printNum(n-1); // n - 1 = 1 //ca
    printf("%d ", n);
}

```

```

void printNum(int n) // recursive
{
    //base case
    if (n==0) //n=1
    {
        return;
    }
    printNum(n-1); // n - 1 = 0 | /
    printf("%d ", n);
}

```

```
void printNum(int n) /
{
    //base case
    if (n==0) //n=0
    {
        return;
    }
    printNum(n-1); // r
    printf("%d ", n);
}
```

```
int fact(int n) // n =4
{
    //base case
    if( n == 0 )
    {
        return 1;
    }

    else
    {
        return n*fact(n-1); // 4 * fact(3)
    }
}
```

```
int fact(int n) // n =3
{
    //base case
    if( n == 0 )
    {
        return 1;
    }

    else
    {
        return n*fact(n-1); // 3 * fact(2)
    }
}
```

```
int fact(int n) // n =2
{
    //base case
    if( n == 0 )
    {
        return 1;
    }

    else
    {
        return n*fact(n-1); // 2 * fact(1)
    }
}
```

```
int fact(int n) // n =1
{
    //base case
    if( n == 0 )
    {
        return 1;
    }

    else
    {
        return n*fact(n-1); // 1 * fact(0)
    }
}
```

```
int fact (int n) // n =0
{
    //base case
    if ( n == 0 )
    {
        return 1;
    }

    else
    {
        return n*fact (n-1) ; |
    }
}
```