

Recursion: (Recursive function) (or) self calling function

→ A Fun. which call itself is known as a Recursive function.

Syntax:

```
void recursion()
```

```
{ recursion(); /* Fun. calls itself */
```

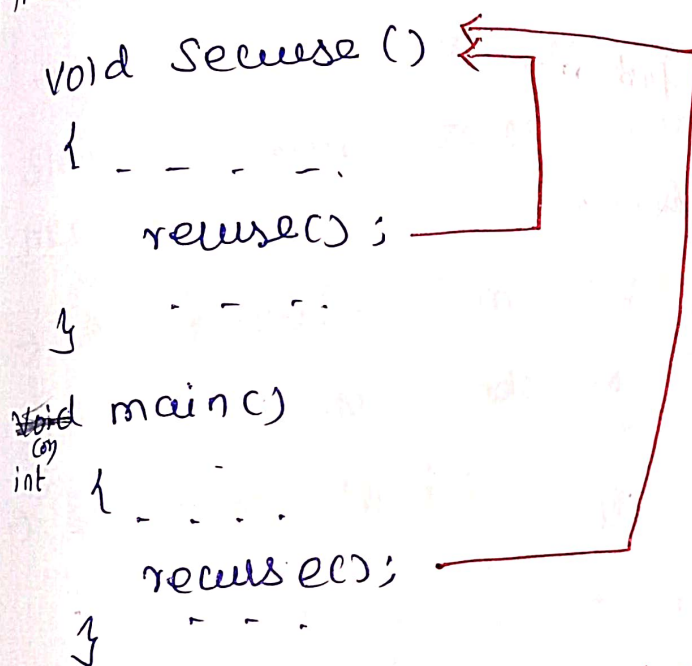
```
}
```

```
int main()
```

```
{ recursion();
```

```
}
```

The recursion works as follows:



Ex: Factorial of 'numbers':

```
#include <stdio.h>
```

```
int fact(int); // Fun. Declaration.
```

```
main()
```

```
{
```

```
int n=4;
```

```
int answer = fact(n); // Fun. call
```

```
printf("The factorial of a number is %d", answer);
```

```
}
```

int fact (int n) // Fun definition.

```

{
    if (n == 1)
    {
        return 1;
    }
    else
    {
        return n * fact(n-1);
    }
}

```

$$\begin{aligned}
 4! &= 4 \times 3 \\
 3! &= 3 \times 2 \times 1 \\
 2! &= 2 \times 1 \\
 1! &= 1
 \end{aligned}$$

O/p:

The factorial of a number is :

Factorial of a Number:

Formula: $\text{fact}(n) = n \times \text{fact}(n-1)$

eg: $\text{fact}(4) = 4 \times \text{fact}(3)$ Here $n=4$.

Recursion working parts.

$$\text{Fact}(4) = 4 \times \text{fact}(3) \quad 4 \times 6 = 24$$

$$\text{fact}(3) = 3 \times \text{fact}(2) \quad 3 \times 2 = 6$$

$$\text{fact}(2) = 2 \times \text{fact}(1) \quad 2 \times 1 = 2$$

$$\text{Fact}(1) = 1$$

①