C Programming

Recursion

Recursion

- **A function that calls itself is known as a recursive function. And, this technique is known as recursion.**
- **The process continues until a condition is met.**

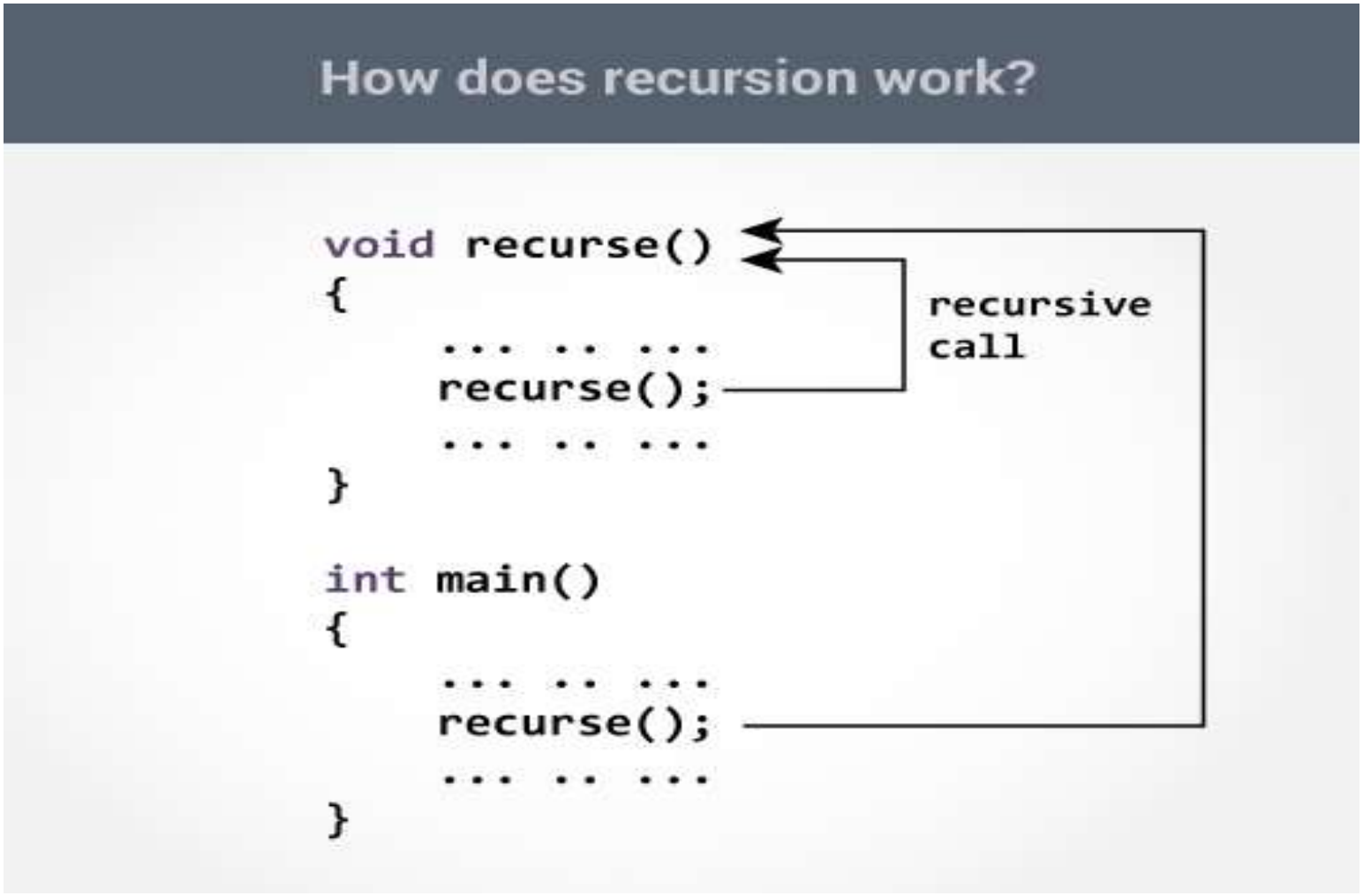
How recursion works?

```
void recurse()
{
    … .. …
    recurse();
    … .. …
}
int main()
{
    … .. …
    recurse();
    … .. …
}
```

Recursion

- **To prevent infinite recursion, if...else statement (or similar approach) can be used where one branch makes the recursive call and other doesn't.**

How Recursion Works?



Recursion: Factorial Example

```
#include <stdio.h>
long int Fact(int n);
int main()
{
   int n;
   printf("Enter a
   positive integer: ");
   scanf("%d", &n);
   printf("Factorial of
   %d = %d", n, Fact(n));
   return 0;
}
```

```
long int Fact(int n)
{
   if (n >= 1)
      return n*Fact(n-1);
   else
      return 1;
}
```

Output

Enter a positive integer:3

Factorial of 3 = 6

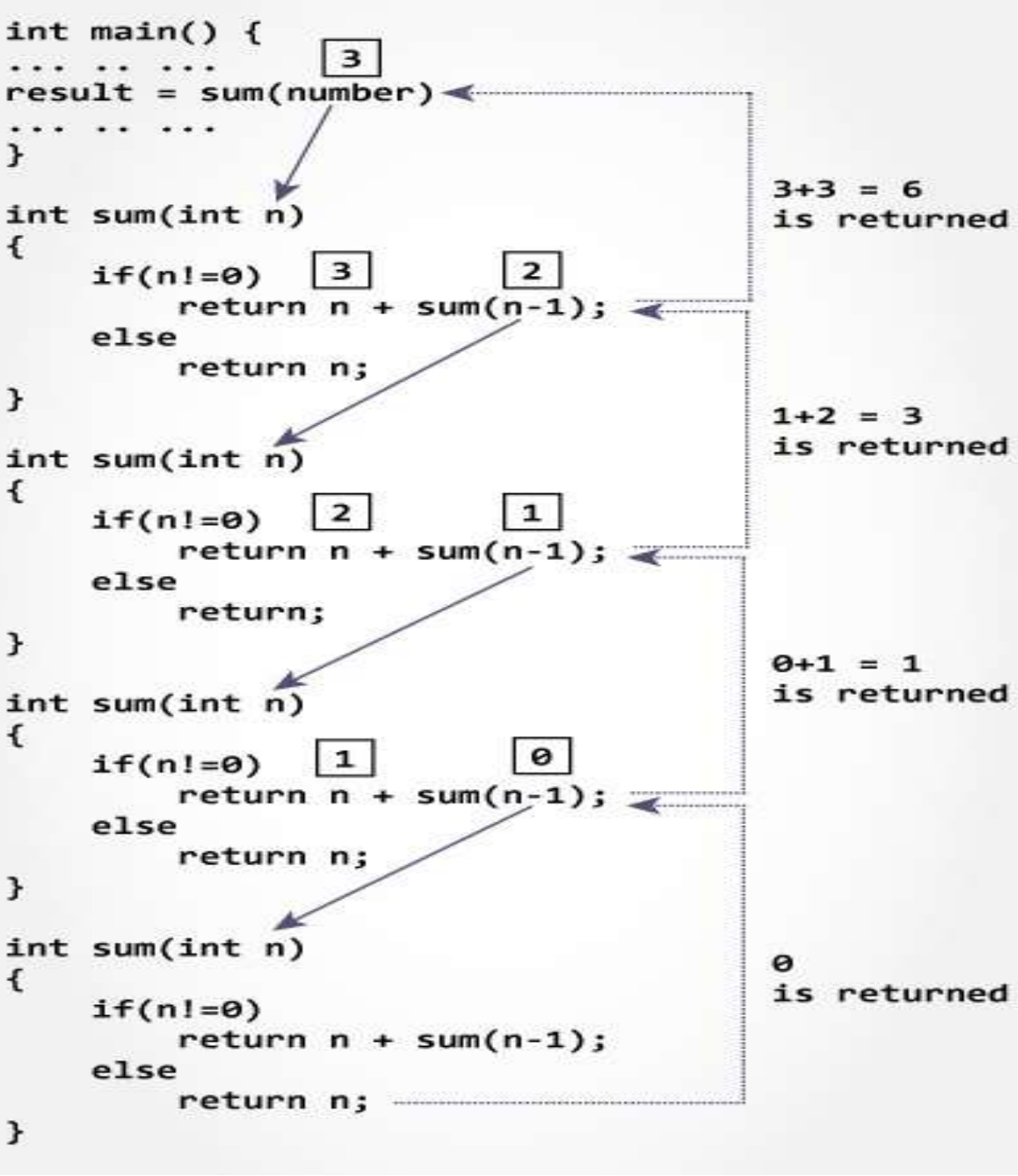
```
main()
{
printf("Factorial of %d = %ld", n,
    Fact(n));
/*suppose n is 3*/
}
long int Fact((3)
{
   if (n >= 1)
      return 3*Fact(2);
/* 3 * 2 is returned */
   else
      return 1;
}
long int Fact(2)
{
   if (n >= 1)
      return 2*Fact(1);
/*return 2 * 1 */
   else
      return 1;
}
long int Fact(1)
{
   if (n >= 1)
      return 1*Fact(0);
/*Returns  1 *1 */
   else
      return 1;
}
long int Fact(int 0)
{
   if (n >= 1)
      return 1*Fact(0);
   else
      return 1;
/*returns 1 */
}
```

Practice Example

WAP to Find sum of numbers from 1 to n using recursion

Solution

```
#include <stdio.h>
int sum(int n);
int main()
{
    int number, result;
    printf("Enter a positive integer: ");
    scanf("%d", &number);
    result = sum(number);
    printf("sum = %d", result);
    return 0;
}
int sum(int num)
{
    if (num!=0)
        return num + sum(num-1); // sum() function calls itself
    else
        return num;
}
```


int main() {
... .
result = sum(number)
... .
}
3
int sum(int n)
{
if(n!=0)
return n + sum(n-1);
else
return n;
}
3
2
3+3 = 6
is returned
int sum(int n)
{
if(n!=0)
return n + sum(n-1);
else
return;
}
2
1
1+2 = 3
is returned
int sum(int n)
{
if(n!=0)
return n + sum(n-1);
else
return n;
}
1
0
0+1 = 1
is returned
int sum(int n)
{
if(n!=0)
return n + sum(n-1);
else
return n;
}
0
is returned

Advantages and Disadvantages

- Recursion makes program elegant and more readable. However, if performance is vital then, use loops instead as recursion is usually much slower.
- **Recursion Vs Iteration?** Need performance, use loops, however, code might look ugly and hard to read sometimes. Need more elegant and readable code, use recursion, however, you are sacrificing some performance.

Some Problems on Recursion

- Write a program in C to Print Fibonacci Series using recursion

0 1 1 2 3 5 8 13

- Write a program in C to find the sum of digits of a number using recursion