Recursion

#cisfun


WE NEED TO GO
DEEPER
troll-face.ru

Iterative programming

Iteration in computing is the repetition of a block of statements within a computer program (using a loop).

Recursion

Recursion is a method where the solution to a problem depends on solutions to smaller instances of the same problem.

A recursive function is a function that calls itself.

Factorial!

#cisfun

Iteration

5! = 5 * 4 * 3 * 2 * 1

5! = 5 * 4 * 3 * 2

5! = 5 * 4 * 6

5! = 5 * 24

5! = 120

Iterative

```c
#include <stdio.h>

int factorial(int n)
{
	int res;
	int i;

	res = 1;
	i = 1;
	while (i <= n)
	{
		res = res * i;
		i++;
	}
	return (res);
}

int main(void)
{
	int f;

	f = factorial(5);
	printf("5! = %d\n", f);
	return (0);
}
```

Recursion

5! = 5 * 4!

4! = 4 * 3!

3! = 3 * 2!

2! = 2 * 1!

1! = 1 * 0!

0! = 1

!n = n * !(n - 1)

factorial(n) = n * factorial(n - 1);

```c
#include <stdio.h>

int factorial(int n)
{
	if (n == 0)
	{
		return (1);
	}
	return (n * factorial(n - 1));
}

int main(void)
{
	int f;

	f = factorial(5);
	printf("5! = %d\n", f);
	return (0);
}
```

Print alphabet

#cisfun

Iterative

Using a loop to iterate through all the letters.

```c
#include <unistd.h>

void print_char(char c)
{
  write(1, &c, 1);
}

void print_alphabet(void)
{
  char c;

  c = 'a';
  while (c <= 'z')
  {
    print_char(c);
    c = c + 1;
  }
}

int main(void)
{
  print_alphabet();
  print_char('\n');
  return (0);
}
```

Recursion

```c
#include <unistd.h>

void print_char(char c)
{
	write(1, &c, 1);
}

void print_all_letters_starting_from(char c)
{
	if (c > 'z')
	{
		return;
	}
	print_char(c);
	print_all_letters_starting_from(c + 1);
}

void print_alphabet(void)
{
	print_all_letters_starting_from('a');
}

int main(void)
{
	print_alphabet();
	print_char('\n');
	return (0);
}
```