Hack `postgres` Source Code: Vol I

Chapter 1 : C & Rust

1.1 Printing format specifier in `postgres --help` in `main.c` in `postgres`

Pi leaned on his chair looking at Go who sat in opposite chair. There is a round table between them at the corner of the restaurant in which they reserved thier dinner date.

Some guy approached them in polite way to them an unpleasant thing.

"Sir, you order might take more than an hour" said the guy who is in restaurant clothes.

"Ok" said Pi as if he experienced same thing thousands of times in the restaurant.

Pi wanted to spent as much time as possible looking at Go and Go was feeling the same. They were not bothered about the delay.

Pi and Go were talking and teasing each other with their words now and then.

After some time, Go asked Pi to write something about format specifiers on the tisse which is available in the form of pile on round table.

"What?" asked Pi confusingly.

"I wanted to know something about format specifiers." said Pi as if it was normal asking such things in restaurant.

"Ok" said Pi baffled by Go's mood swings.

Pi took up a tissue and tried to write something..

On Tissue -

C:

```
#include<stdio.h>

int
main() {

    // Integer format specifier
    short int  si = 2; // 2 byte int
    printf("%hd", si);
    int i = 4; // 4 byte int
    printf ("%d", i);

    long int li = 8; // 8 byte int
    printf("%ld", li);

    // Float format specifier
    float f = 4; // 4 byte fractional number
    printf("%f", f);

    double d = 8; // 8 byter fractional number
    printf("%f", d);

    //Character format specifier
    char c = '1'; // 1 byte character
    printf("%c", c);

    // String format specifier
    // There is no type `string` in C.
    // But, there is format specifer for as we have for primitive types.
    char c = '1';
    printf("%s", c);

    return 0;
}
```

Rust:

```
fn main() {

    //Integer format specifiers
    let _i8: i8 = 1; // 1 byte
    let _i16: i16 = 2; // 2 byte
    let _i32: i32 = 4; // 4 byte
    let _i64: i64 = 8; // 8 byte
    let _128: i128 = 16; // 16 byte

    println!("{}{}{}{}{}", _i8, _i16, _i32, _i64, _i128);

    //Float format specifiers
    let _f32: f32 = 4.0; // 4 bytes
    let _f64: f64 = 8.0; // 8 bytes

    println!("{}{}", _f32, _f64);

    //Character format specifiers
    let _char: char = '1'; // 4 bytes

    println!("{}", _char);

    //Bool format specifier
    let _bool: bool = true; // 1 byte

    println!("{}", _bool);

}
```

Observations:

1. There are different format specifiers for different data types in C. But, we don't have such thing in Rust.
2. There is bool in C from `stdbool.h` but there is no format specifier for it.
3. Strings in C & Rust are not primitive types.
4. There is implicit conversion to float in C when we assign integer. But, in Rust, compiler insists us to assign only floats to float types.

After completion of the writing, Pi handed over tissue while caressing her smooth hand.

Meanwhile the guy in white clother came to their table to serve their ordered food. They had great meal and exchange of looks and thoughts while having dinner.

Go and Pi reached underground parking lot in the restaurant. She was approaching the car and he was following her with less distance and enjoying her from back. She approached the car door and turned back to ask him car keys. As she was moving towards him, he immediatly pushed her against car door in the parking lot's dim light. He was looking straight into her.

"What are you doing?" asked Go gasping.

In the shadows her face was so close to him that he could smell the sweet ropy fragrance of the sherry she had been drinking. She blossomed like a flower at his lip's touch. Her lips were soft, almost silken and pillowy against his own. He could feel the soft tickel of her breath beneath his nose, fingers carding through her hair as they breathed each other in.