Hack `postgres` Source Code: Vol I

Chapter 1 : C & Rust

1.1 Arrays in `pg_config.c` in `postgres`

Note: Array `info_items` is initialized with structure as type in `pg_config.c` . It is modified for primitive type in our case.

Cut to - White board

C:

```c
#include<stdio.h>

int
main() {

    /*
    * Arrays are used to store "fixed length" values of "same type".
    * Values in array are stored in contiguous memory locations in RAM
    * Array values are retrieved by index notation.
    */
    int _binary[2] = {1,0};
    printf("%d\n", _binary[0]); // index 0 is to access first element
    printf("%d\n", _binary[1]); // index 1 is to access second element

    return 0;

}
```

Rust:

```rust
fn main() {

    /*
    * Rust's array's type and size are defined in a bracket separated
    * by a semicolon. Array values are stored in bracket instead of
    * curly brackets.
    */
    let _binary: [i32; 2] = [1,0];

    println!("{} {}", _binary[0], _binary[1]);

}
```

Observations:

1. Arrays in Rust and C follows same functionality. Both allows only fixed length which means the size of array is not modified once it is initialized. Both allows only same type of elements. Both follows same index notation.
2. But, the rust's way of initialization syntax is different from C.