Hack postgres Source Code: Vol I

Chapter 1 : C & Rust

1.39 Dangling Pointer - C

What happens if we return an address of automatic variable/local variable from a function to someother function like below -

```
#include<stdio.h>

int* add(int a, int b);

int main() {

    int a=10, b=10, *res;

    res = add(a,b);

    return 0;
}

int* add(int a, int b) {

    int sum = a + b;

    return &sum;
}
```

C throws an error during compilation.

But Why?

It is obvious that it throws an error. Because, function calls are stored in stack memory one above another (It was discussed in some previous chapters). As soon as function call is finished, all the stack memory allocated to that function specific variables gets destroyed/cleaned up.

In our case, we are sending an address of variable which is going to get destroyed as soon as `add` function call is completed. And C is not ready to make its pointers point such memory locations. If we do so, people use the word `dangling pointer` to label the pointer involved in those kind of cases.

So.. here dangling pointer is `res` ..??

Yeah..