# Call-stack (or run-time stack).

How are function calls managed/implemented?

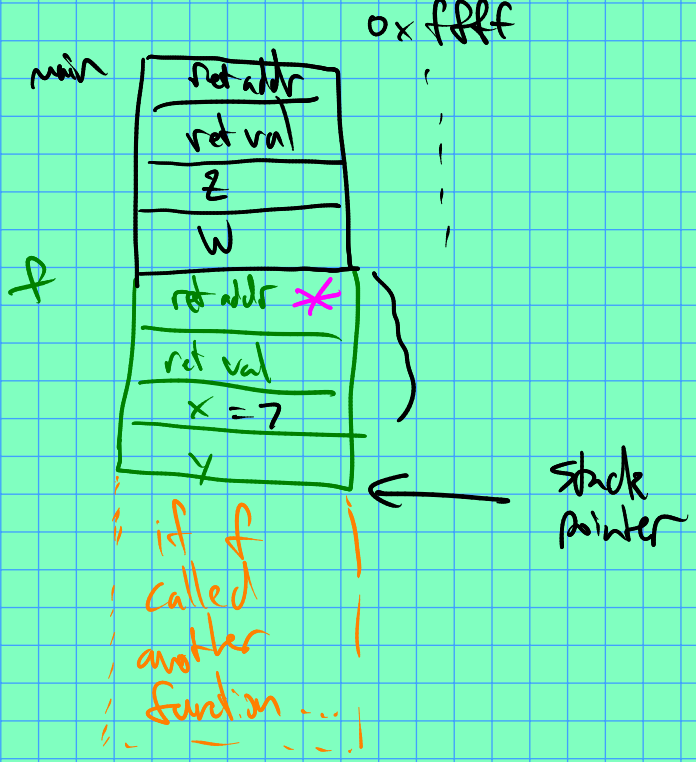
```
int f(int x) {
    int y;
    ...
    return aa;
}

int main() {
    int z = 7;
    int w = f(z);
    ...
}
```

What data needs to be stored during a call to f?

- return address (which instruction to do after the call)
- storage for parameters (like x in f(...))
- local variables (y in f(...))
- storage for return value

All this data → is managed on a stack.

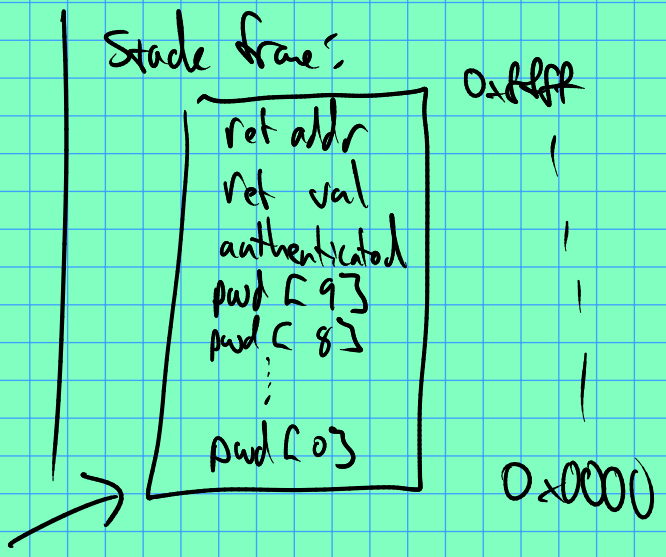
Here's a picture for main calling f:




Homework: read about "calling conventions"

Sometimes these behind the scenes details actually matter to the functionality of your C++ program!!

```
pwdCheck() {
    int authenticated = 0;
    char pwd[10];
    :
}
```




authenticated ≈ pwd[10]

A[i] ≡ *(A+i)