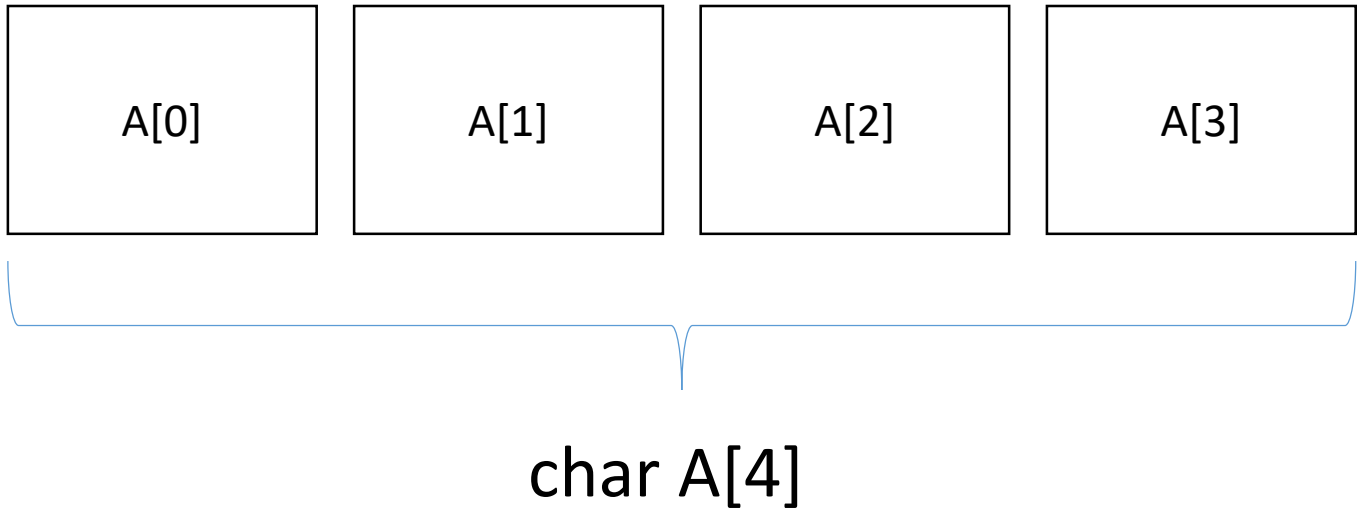
‘C’ Reversing

nafiez

Basic to brother!

Data Types	Bytes
char	1 byte
short	2 bytes
int	4 bytes (platform word)
long	4 bytes
float	4 bytes floating point
double	8 bytes floating point

Arrays



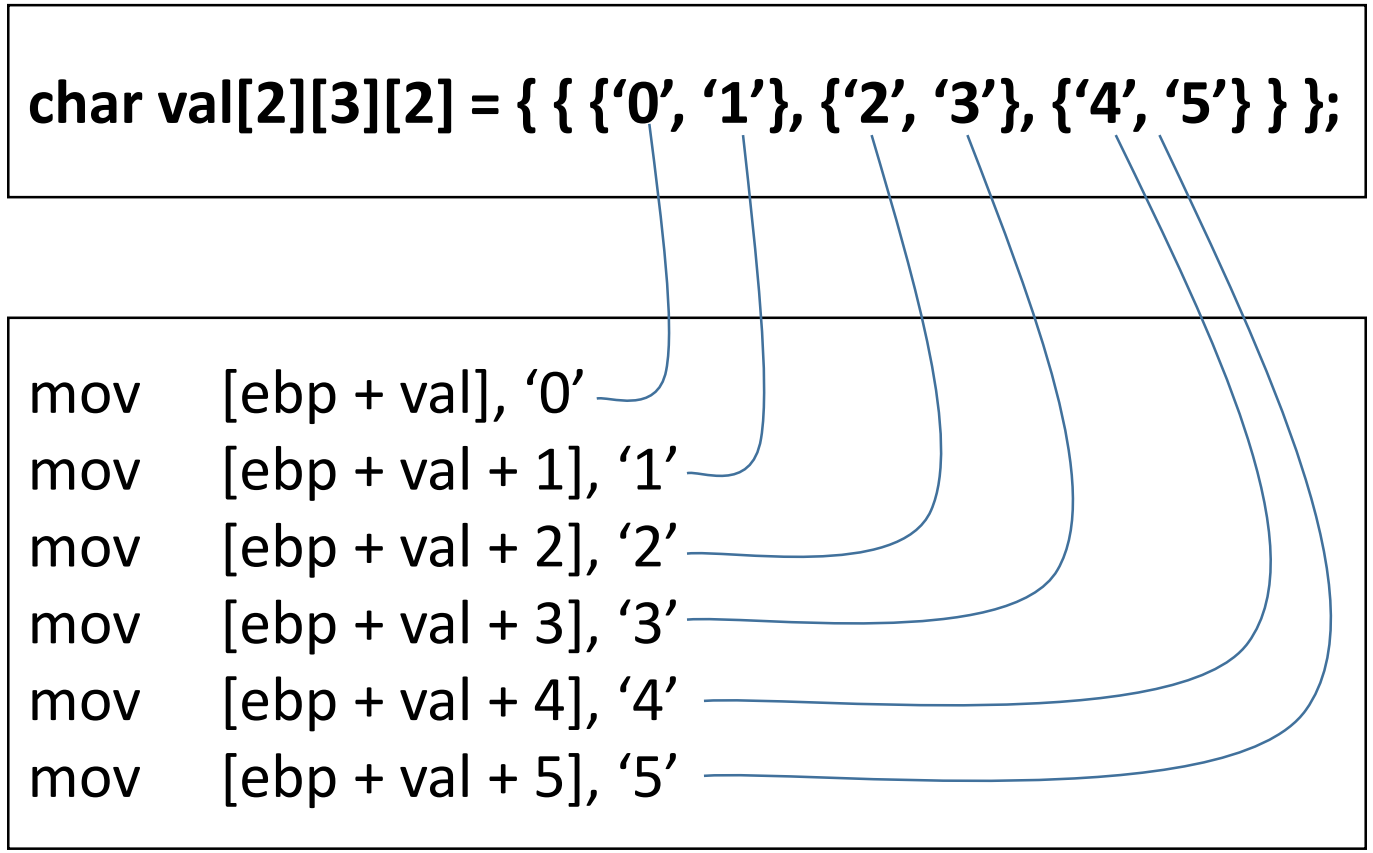
Example

C code

```
char val[2][3][2] = { { {'0', '1'}, {'2', '3'}, {'4', '5'} } };
```

Assembly

```
mov     [ebp + val], '0'
mov     [ebp + val + 1], '1'
mov     [ebp + val + 2], '2'
mov     [ebp + val + 3], '3'
mov     [ebp + val + 4], '4'
mov     [ebp + val + 5], '5'
```

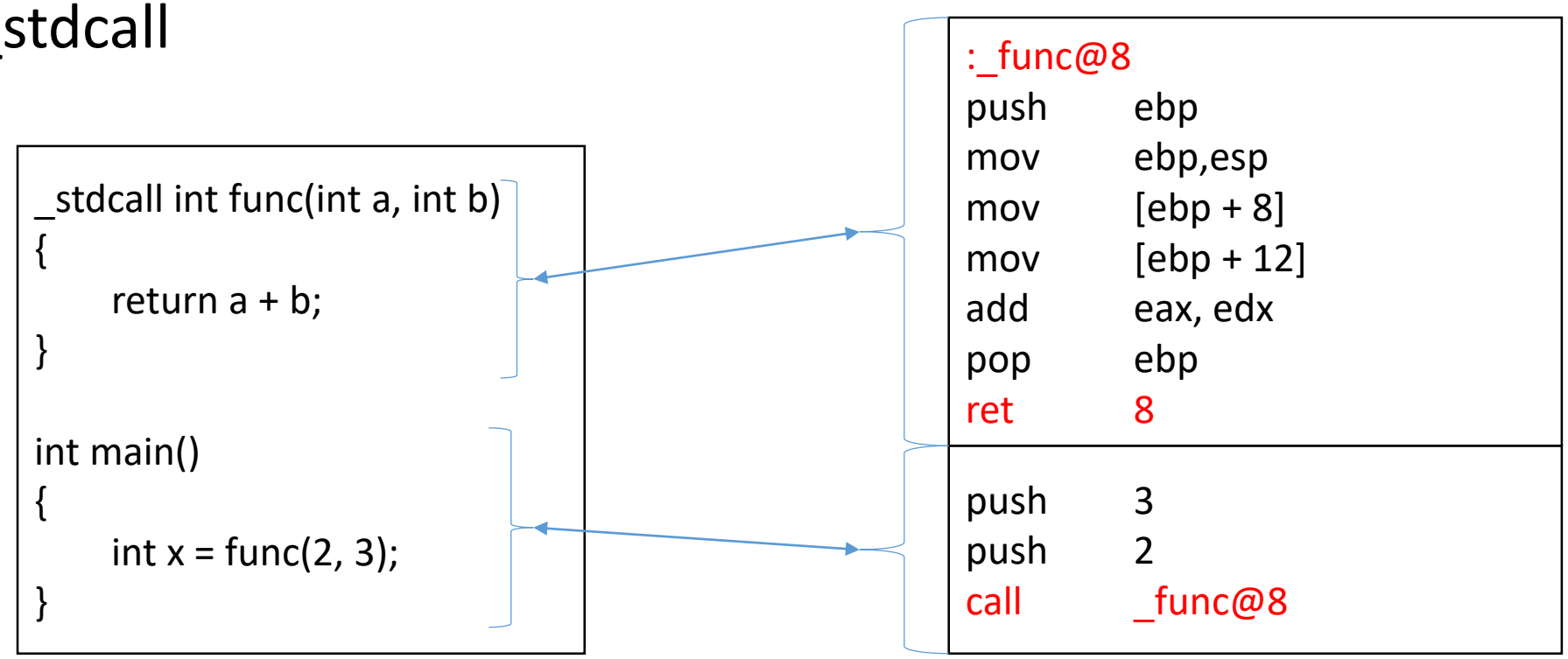


Calling Conventions

- Methods for function to be implemented and called by machine
- Specify how arguments are passed to a function
- Basically its about specifying how a function call in C / C++ converted into assembly.
- In Visual Studio, there few types of calling conventions (we focused on x86 processors):
  - STDCALL
  - CDECL
  - FASTCALL

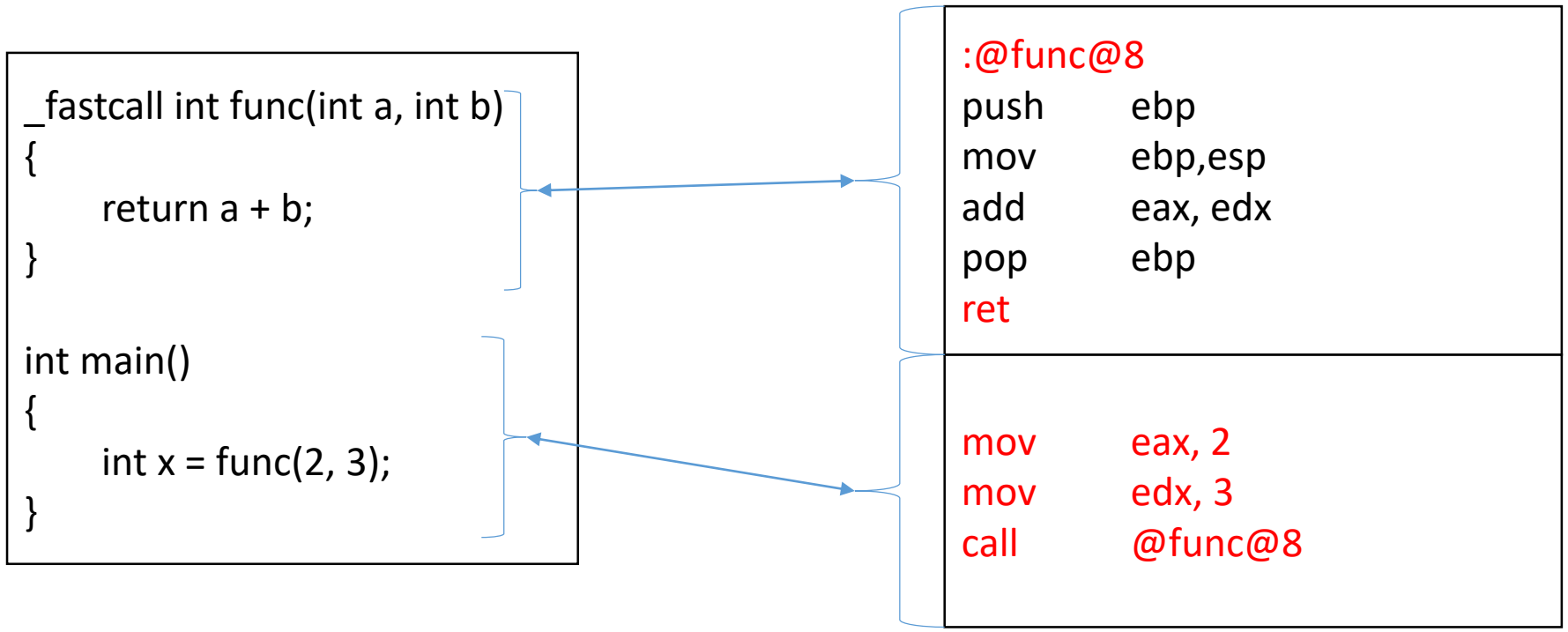
STDCALL

- Used to call WIN32 API functions
- __stdcall



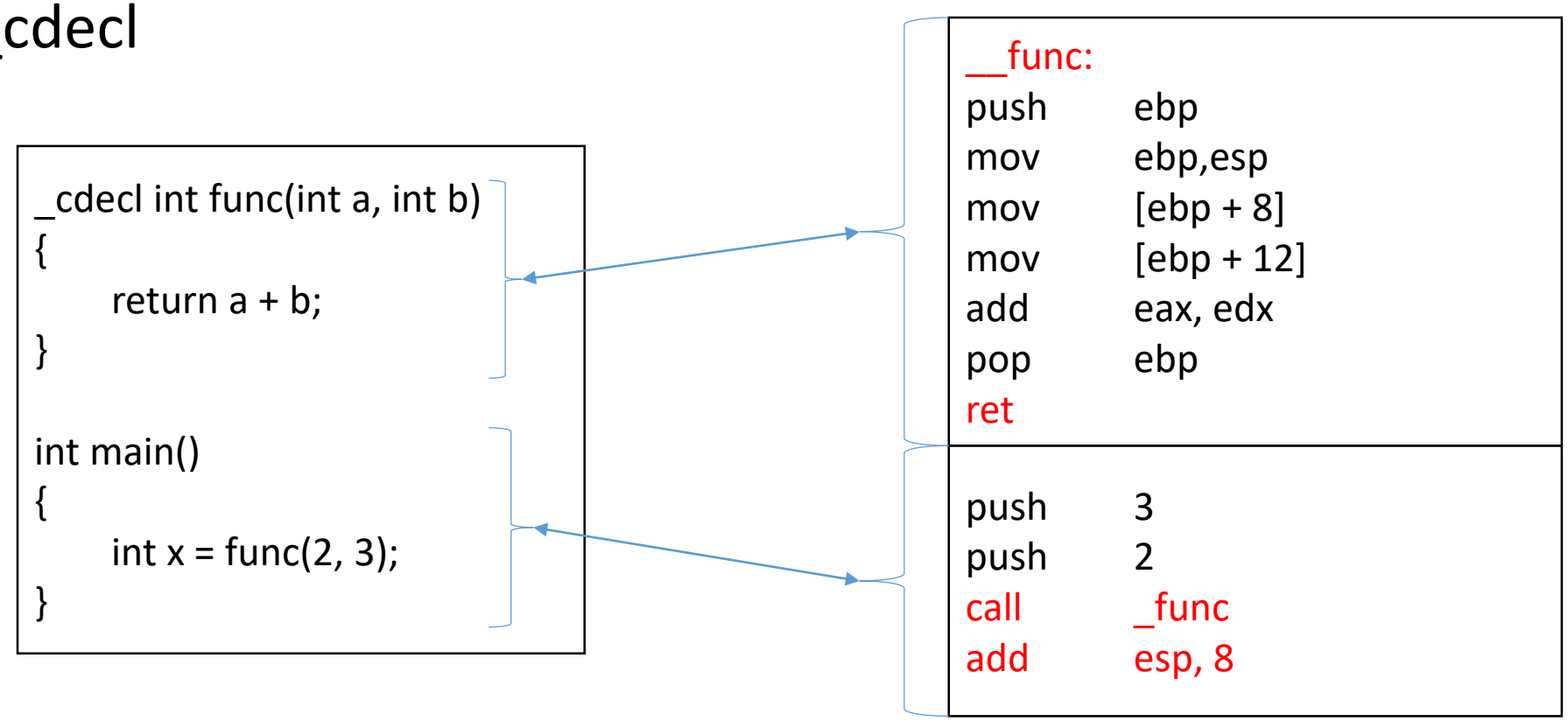
FASTCALL

- Specifies arguments to functions are to be passed in registers
- __fastcall



CDECL

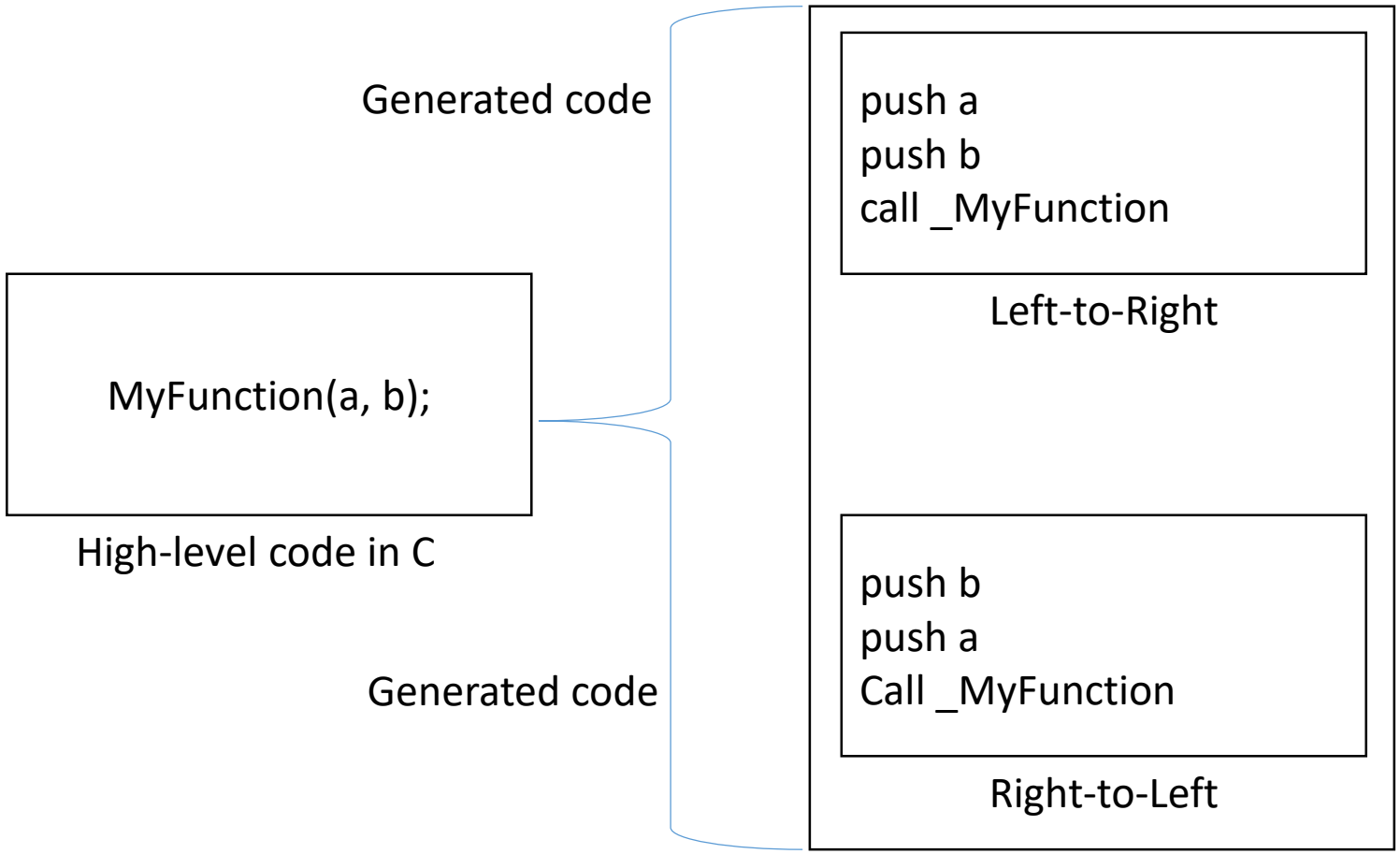
- Default calling convention for C / C++ programs
- __cdecl



Passing Arguments

- Calling function writing to data in place where the called will look for them.
- Arguments are passed before *call* instruction is executed.
- Right-to-Left (R2L) and Left-to-Right (L2R)
  - Arguments are passed in subroutine

Example: R2L & L2R



Return Value

- If functions returns a value, it will reliably received by function's caller.
- Called function stores the return value before executing *ret* instruction.

Stack Cleaning (Caller-Callee)

- Arguments push onto stack, and it will need to pop out from the stack.
- Caller-Callee responsible to do cleaning in the stack (reset the stack pointer) to eliminate passed arguments.
- Caller
  - Parent function calls subroutine. Execution resumes in the calling function after subroutine call, unless it terminated inside the subroutine
- Callee
  - child function called by parent