

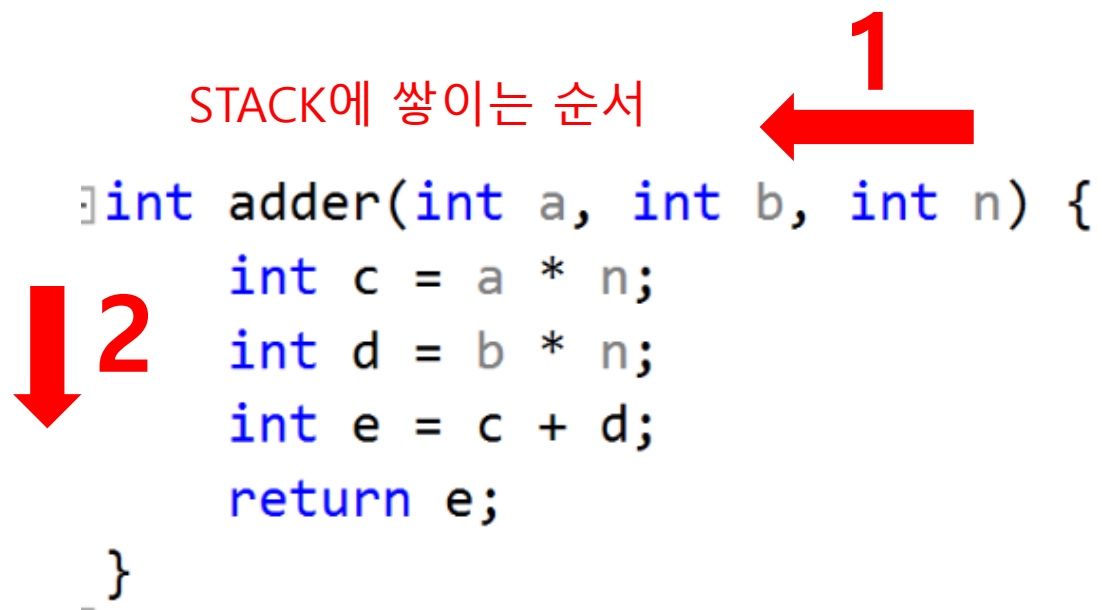
Stack frame

```
int adder(int a, int b, int n) {  
    int c = a * n;  
    int d = b * n;  
    int e = c + d;  
    return e;  
}
```

```
int main(void) {  
    int a = 10;  
    int b = 20;  
    int n = 2;  
    int res = adder(a, b, n);  
    return 0;  
}
```

Stack frame

STACK에 쌓이는 순서



```
int adder(int a, int b, int n) {  
    int c = a * n;  
    int d = b * n;  
    int e = c + d;  
    return e;  
}
```

Stack frame

```
14:    int res = adder(a, b, n);
```

00381723	mov	eax,dword ptr [n]
00381726	push	eax
00381727	mov	ecx,dword ptr [b]
0038172A	push	ecx
0038172B	mov	edx,dword ptr [a]
0038172E	push	edx

stack pointer가 가리키는 곳에 push

esp : extended stack pointer

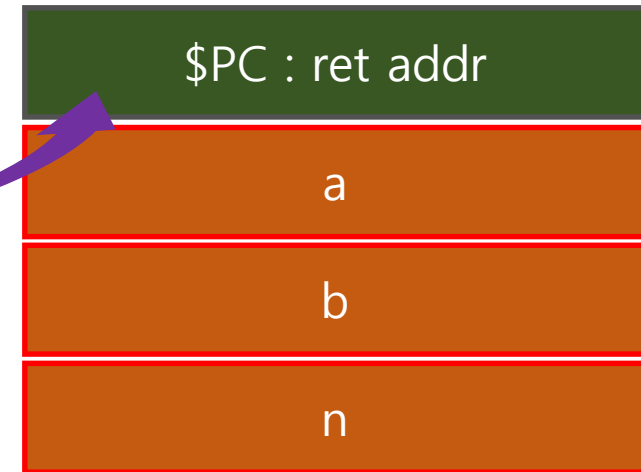
esp →

n

Stack frame

```
14:      int res = adder(a, b, n);  
00381723  mov          eax,dword ptr [n]  
00381726  push         eax  
00381727  mov          ecx,dword ptr [b]  
0038172A  push         ecx  
0038172B  mov          edx,dword ptr [a]  
0038172E  push         edx  
0038172F  call 함수 호출_adder (0381109h)  
00381734  add          esp,0Ch
```

esp →



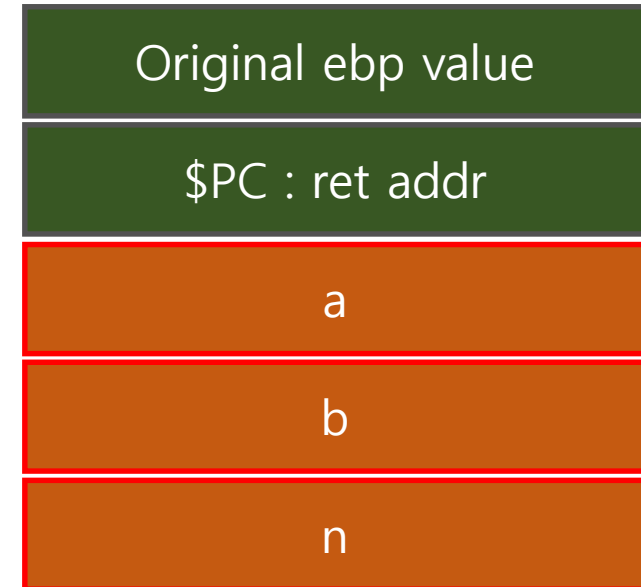
Stack frame

```
3: int adder(int a, int b, int n) {
```

```
00381690  push      ebp      ≤ 1ms elapsed
00381691  mov       ebp,esp
00381693  sub       esp,0E4h
```

ebp : extended **base pointer**

ebp = esp →



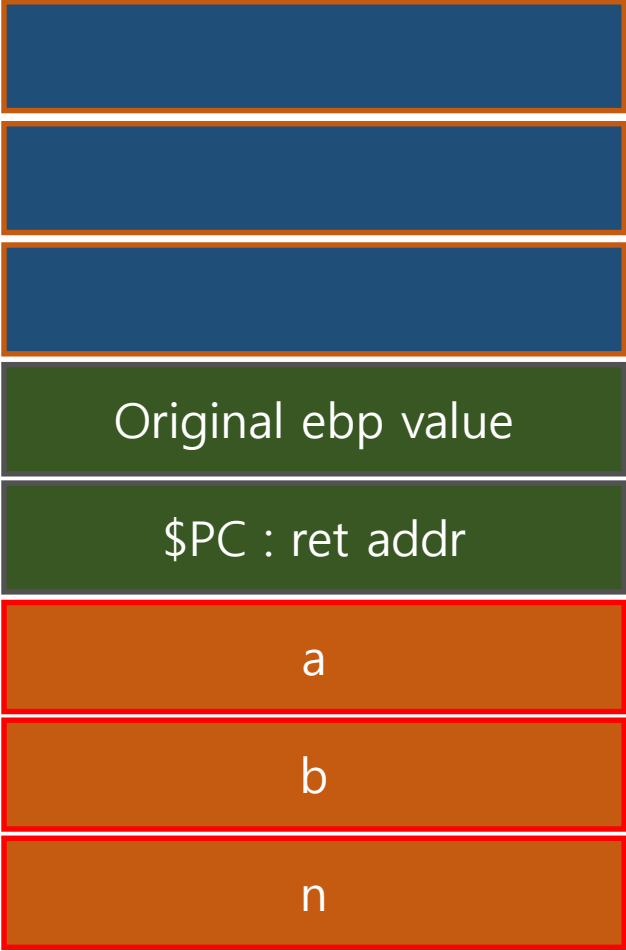
Stack frame

```
3: int adder(int a, int b, int n) {  
00381690  push      ebp      ≤ 1ms elapsed  
00381691  mov       ebp,esp  
00381693  sub       esp,0E4h
```

지역 변수 공간 미리 확보

esp →

ebp →



Stack frame

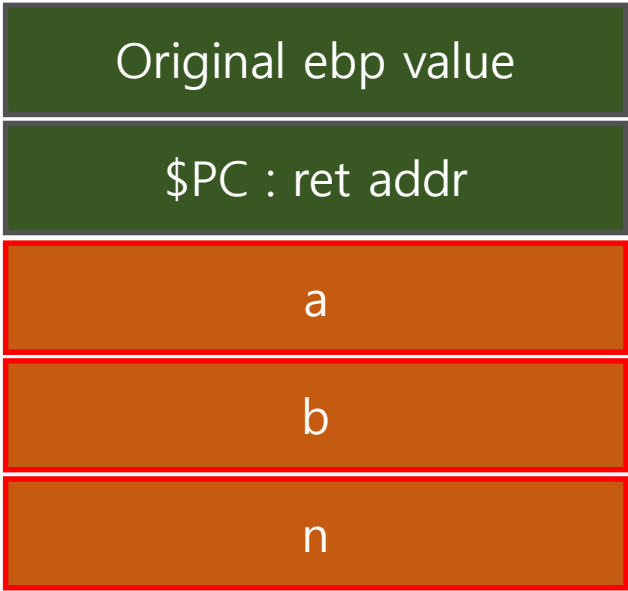
```

    7:      return e;
003816CB  mov      eax,dword ptr [e]
    8:  }
003816CE  pop      edi
003816CF  pop      esi
003816D0  pop      ebx
003816D1  mov      esp,ebp
003816D3  pop      ebp
003816D4  ret

```

실제 지우지는 않지만 stack pointer가 움직인 것이
결국엔 해제

esp = ebp →



```
    7:    return e;
003816CB  mov     eax,dword ptr [e]
    8:  }
003816CE  pop     edi
003816CF  pop     esi
003816D0  pop     ebx
003816D1  mov     esp,ebp
003816D3  pop     ebp
003816D4  ret
```

CPU

ebp

Original ebp value

eip

\$PC : ret addr

eip : extended **instruction pointer**

esp



a

b

n


```
14:      int res = adder(a, b, n);
00381723  mov     eax,dword ptr [n]
00381726  push    eax
00381727  mov     ecx,dword ptr [b]
0038172A  push    ecx
0038172B  mov     edx,dword ptr [a]
0038172E  push    edx
0038172F  call    _adder (0381109h)
00381734  add     esp,0Ch
```

