함수 호출 시의 Stack 모습

1 CM 코드

```
int sum;                        // global variable

void add(int i, int j) {        // parameters = local variables
  sum = i + j;
}

void main(void) {
  int x;                        // local variables
  int y;

  input x;                      // Keyboard Input
  input y;
  add(x, y);                    // function call
  output sum;                   // print global variables
}
```

2 TM code

```
// ======== Startup ==================================
0000: LD   gp, 0(0)     // gp = dMem[0]
0001: ST   0, 0(0)     // dMem[0] = 0

0002: LDA  fp, -1(gp)   // fp = gp + sizeof(global vars)
0003: LDA  sp, -1(gp)   // sp = gp + sizeof(global vars)

0004: PUSH fp         // push fp
0005: LDA  0, 2(pc)     // r0 = return addr
0006: PUSH 0         // push r0
0007: LDC  pc, 17       // jump to main

0008: HALT // halt

// ======== add(int i, int j) ======================
0009: LDA  sp, 0(sp)   // for local vars

0010: LD   0, -2(fp)   // r0 = i
0011: LD   1, -3(fp)   // r1 = j
0012: ADD  0, 0, 1     // r0 = r0 + r1
0013: ST   0, 0(gp)     // sum = r0

0014: LDA  sp, 0(fp)   // sp = fp
0015: LD   fp, 0(fp)   // fp = dMem[fp]
0016: LD   pc, -1(sp)   // pc = dMem[sp-1]
```

```
// ========== main() ================================
0017: LDA  sp, -2(sp)       // for local vars

0018: IN   0                // r0 <= in
0019: ST   0, -2(fp)       // x = r0
0020: IN   0                // r0 <= in
0021: ST   0, -3(fp)       // y = r0

0022: LDA  sp, -2(sp)       // space for oldfp and return addr
0023: LD   0, -2(fp)       // r0 = x
0024: PUSH 0               // push r0, i
0025: LD   0, -3(fp)       // r0 = y
0026: PUSH 0               // push r0, j

0027: ST   fp, -4(fp)       // save oldfp
0028: LDA  fp, -4(fp)       // set new fp
0029: LDA  0, 2(pc)         // r0 = return addr
0030: ST   0, -1(fp)       // dMem[fp-1] = r0
0031: LDC  pc, 9            // jump to add

0032: LD   0, 0(gp)         // r0 = sum
0033: OUT  0             // r0 => out

0034: LDA  sp, 0(fp)       // sp = fp
0035: LD   fp, 0(fp)       // fp = dMem[fp]
0036: LD   pc, -1(sp)       // pc = dMem[sp-1]
```

3 add 함수 (009 번지) 수행 시의 Stack 모습

...	↑ 하위주소
j (2nd parameter)	← $fp-3$
i (1st parameter)	← $fp-2$
return address (032)	← $fp-1$
fp(main)	← fp(add)
y (지역 변수)	← $fp-3$
x (지역 변수)	← $fp-2$
return address (008)	
fp(startup)	← fp(main)
sum (전역변수)	← gp 최 상위주소