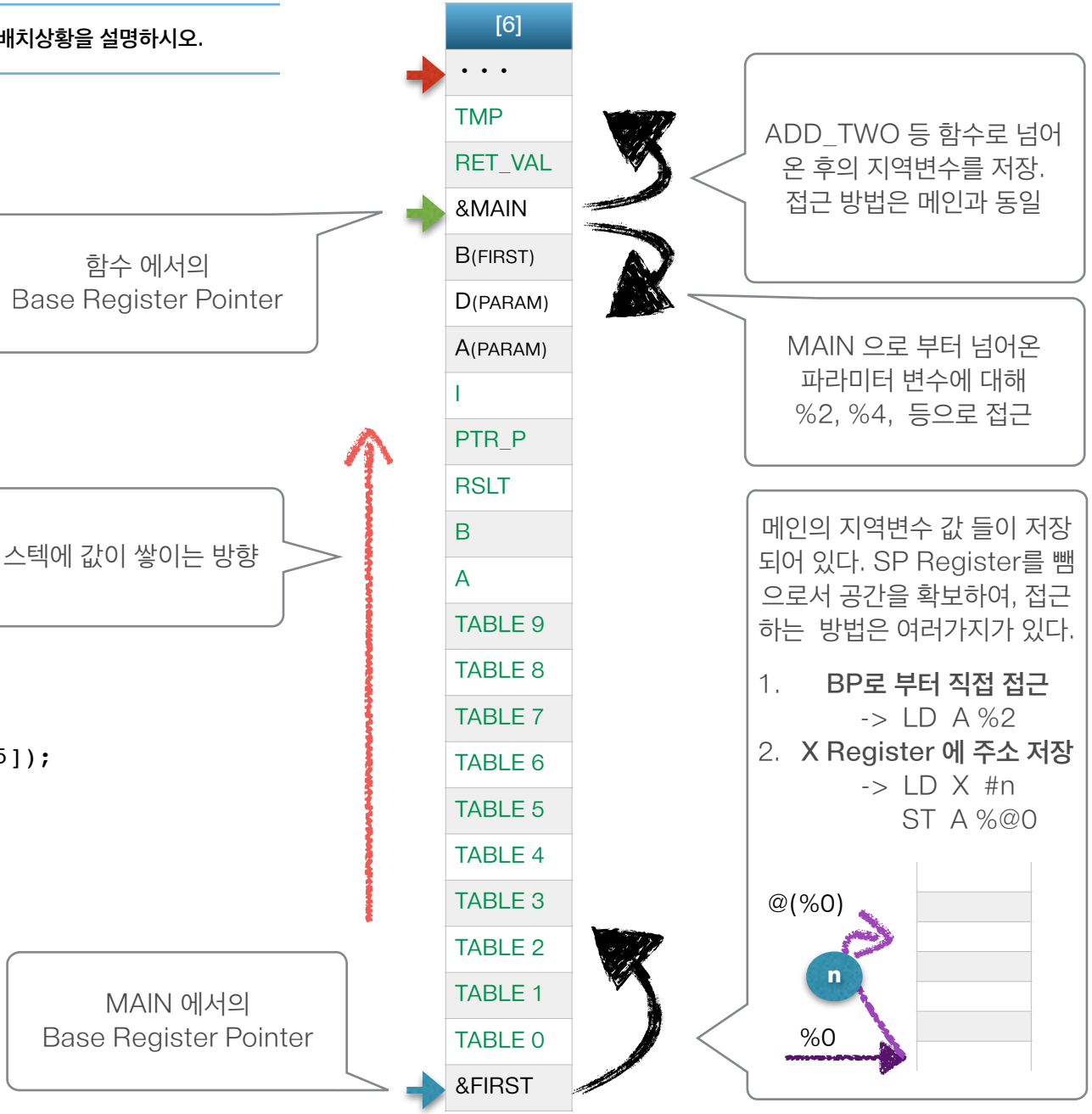
다음 프로그램을 LMC 코드로 작성하고 각 변수들의 배치상황을 설명하시오.

```
static int add_two(int x, int y) {
    int tmp;
    tmp = x + y;
    return tmp;
    }
main() {
    int table[10];
    int a, b, result;
    int *p;
    int i;

    for(i=9; i--; i>=0)
         table[i] = i+1;

    result = add_two(2,3);
    OUT(result,stdout);

    a = table[2];
    b = table[3];
    result = add_two(a,b);
    OUT(result,stdout);

    result = add_two(table[4],table[5]);
    OUT(result,stdout);

    p = table;
    result = add_two(p[1], *(p+2));
    OUT(result,stdout);

    p = &table[5];
    result = add_two(p[1], *(p+2));
    OUT(result,stdout);
}
```



*** 다음은 어셈블리 후 생성된 리스트입니다. 7 페이지에 각 변수 배치 상황을 그림으로 설명하였습니다.**

```
Monitor> asm 4 5 6
Execute% Assemble 4:cassette/myTest1.lmc
                 0000:0000   1 IMPLE_PROG   START   0
                 0000:0000   2 KEYBOARD_DEC EQU     10
                 0000:0000   3 SCREEN_DEC   EQU     12
                 0000:0000   4 SCREEN_TXT   EQU     13
                             5
4355 0404 0000   0000:0000   6 FIRST        LD      SP #STK_BTM
6700 0033 0000   0000:0003   7              CALL    MAIN
0700             0000:0006   8              COB
                             9
4115             0000:0007  10 ADD_TWO      LD      B SP
1655 0002 0000   0000:0008  11              SUB     SP #1*2         // RETURN_VALUE(%-2)
1655 0002 0000   0000:0011  12              SUB     SP #1*2         // LOCAL VARIABLE TMP(%-4)
                            13
9884             0000:0014  14              PUSH    X
9880             0000:0015  15              PUSH    A
                            16
4205 0006        0000:0016  17              LD      A %6
4245 0004        0000:0018  18              LD      X %4
1104             0000:0020  19              ADD     A X
                            20
5205 9996        0000:0021  21              ST      A %-4           // SET TMP
                            22
9890             0000:0023  23              POP     A
9894             0000:0024  24              POP     X
                            25
4151             0000:0025  26              LD      SP      B
                            27
9880             0000:0026  28              PUSH    A
4205 9996        0000:0027  29              LD      A %-4           // GET TMP
5205 9998        0000:0029  30              ST      A %-2           // SET RETURN_VALUE
9890             0000:0031  31              POP     A
```

~[4]

~[5]

~[6]

~[7]

```
                                 32
9999                0000:0032    33                RET
                                 34
4115                0000:0033    35 MAIN           LD              B SP
1655 0020 0000      0000:0034    36                SUB             SP #10*2        // LOCAL ARRAY TABLE(%-2)
1655 0006 0000      0000:0037    37                SUB             SP #3*2         // LOCAL VARIABLE
                                                                                   // A(%-22),B(%-24),RSLT(%-26)
1655 0002 0000      0000:0040    38                SUB             SP #1*2         // LOCAL PTR P(%-28)
1655 0002 0000      0000:0043    39                SUB             SP #1*2         // LOCAL VARIABLE I(%-30)
                                 40
4325 0010 0000      0000:0046    41                LD              C #10
4345 9998 9999      0000:0049    42                LD              X #-2
                                 43
4102                0000:0052    44 FOR_LOOP       LD              A C
5207 0000           0000:0053    45                ST              A %@0
1645 0002 0000      0000:0055    46                SUB             X #2
5990 0052 0000      0000:0058    47                JLOOP   FOR_LOOP
                                 48
0000                0000:0061    49 ADD_1          NOP ////////////////////// add_two(a,b); ////////////////////////
                                 50
4205 9994           0000:0062    51                LD              A %-(2+(2*2))
5205 9978           0000:0064    52                ST              A %-22          // A = TABLE[2]
                                 53
4235 9992           0000:0066    54                LD              D %-(2+(3*2))
5235 9976           0000:0068    55                ST              D %-24          // B = TABLE[3]
                                 56
                                 57
9880                0000:0070    58                PUSH    A
9883                0000:0071    59                PUSH    D                       // PARAMETER
9881                0000:0072    60                PUSH    B                       // B->DATA BACKUP
                                 61
6700 0007 0000      0000:0073    62                CALL    ADD_TWO
                                 63
4235 9998           0000:0076    64                LD              D %-2           // GET RETURN_VALUE(TO D)
9891                0000:0078    65                POP             B
5235 9974           0000:0079    66                ST              D %-26          // RLST = RETURN_VALIE
```

~[1]

~[2]

~[3]

~[8]

~[9]

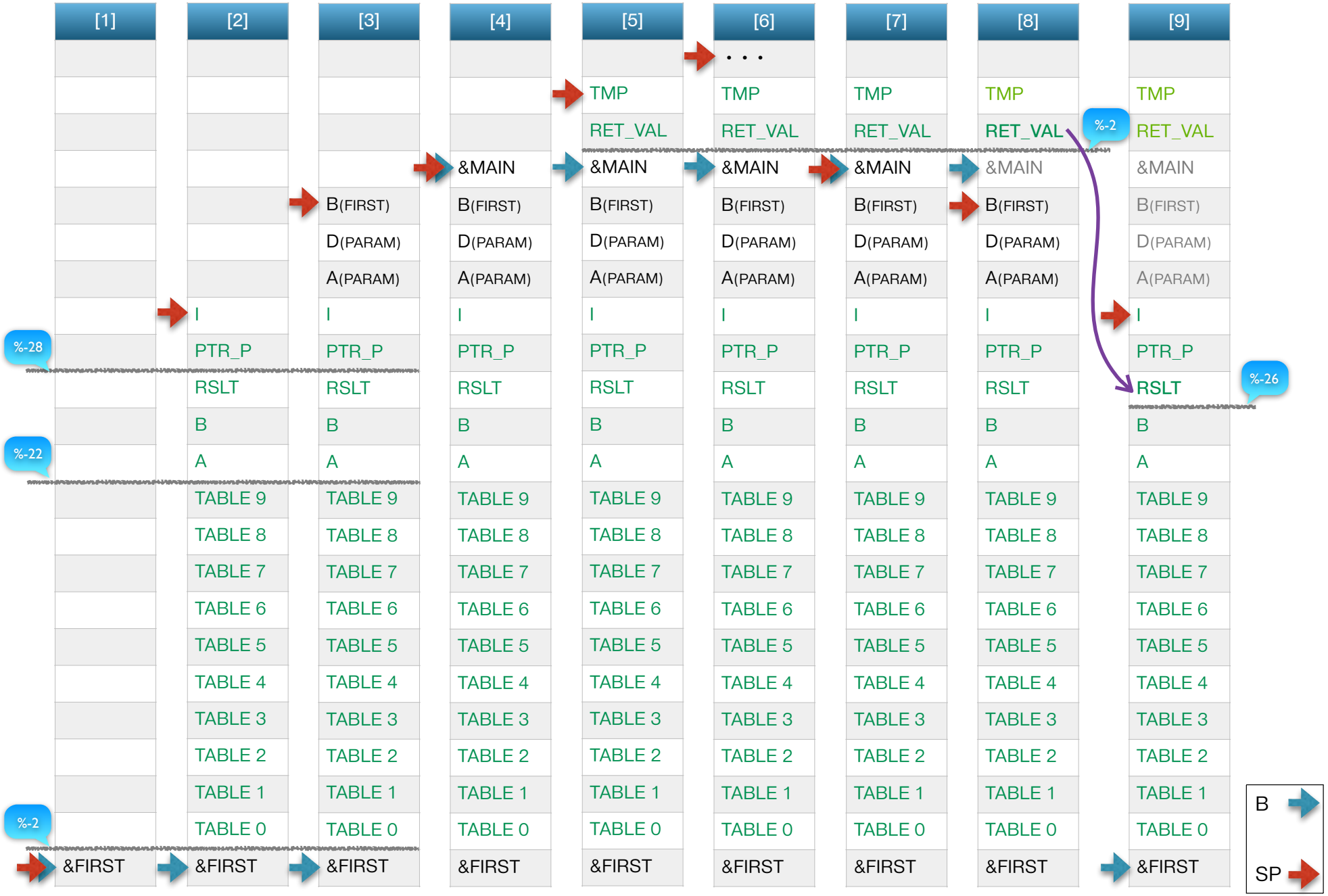
```
1355 0004 0000   0000:0081                ADD             SP #(2*2)       // RESET ST

4205 9974        0000:0084                LD              A %-26
0612             0000:0086                OUT             12 ////////// OUT(RSLT) ////////////////////////

4305 0010 0000   0000:0087                LD              A #10
0613             0000:0090                OUT             SCREEN_TXT

0000             0000:0091  ADD_2         NOP //////// add_two(table[4],table[5]) ////////////////////////

4205 9990        0000:0092                LD              A %-(2+(4*2))
9880             0000:0094                PUSH            A
4205 9988        0000:0095                LD              A %-(2+(5*2))
9880             0000:0097                PUSH            A               // PARAMETER
9881             0000:0098                PUSH            B               // B->DATA BACKUP

6700 0007 0000   0000:0099                CALL            ADD_TWO

4235 9998        0000:0102                LD              D %-2           // GET RETURN_VALUE(TO D)
9891             0000:0104                POP             B
5235 9974        0000:0105                ST              D %-26          // RLST = RETURN_VALIE
1355 0004 0000   0000:0107                ADD             SP #(2*2)       // RESET ST

4205 9974        0000:0110                LD              A %-26
0612             0000:0112                OUT             12 ////////// OUT(RSLT) ////////////////////////

4305 0010 0000   0000:0113                LD              A #10
0613             0000:0116                OUT             SCREEN_TXT

0000             0000:0117  ADD_3         NOP /////////// add_two(p[1], *(p+2)); ////////////////////////

4101             0000:0118                LD              A B
1605 0002 0000   0000:0119                SUB             A #2
5205 9972        0000:0122                ST              A %-28          // PTR_P = #TABLE(%-2)

4245 9972        0000:0124                LD              X %-28          // GET *PTR_P
```

```
1645 0002 0000   0000:0126 103              SUB             X #(1*2)        // = PTR_P[1]
4302 0000 0000   0000:0129 104              LD              A @0
9880             0000:0132 105              PUSH            A
4245 9972        0000:0133 106              LD              X %-28
4302 9996 9999   0000:0135 107              LD              A @(-2*2)       // = *(PTR_P+2)
9880             0000:0138 108              PUSH            A               // PARAMETER
9881             0000:0139 109              PUSH            B               // B->DATA BACKUP
                           110
6700 0007 0000   0000:0140 111              CALL            ADD_TWO
                           112
4235 9998        0000:0143 113              LD              D %-2           // GET RETURN_VALUE(TO D)
9891             0000:0145 114              POP             B
5235 9974        0000:0146 115              ST              D %-26          // RLST = RETURN_VALIE
1355 0004 0000   0000:0148 116              ADD             SP #(2*2)       // RESET ST
                           117
4205 9974        0000:0151 118              LD              A %-26
0612             0000:0153 119              OUT             12 ///////// OUT(RSLT) ////////////////////////
                           120
4305 0010 0000   0000:0154 121              LD              A #10
0613             0000:0157 122              OUT             SCREEN_TXT
                           123
0000             0000:0158 124 ADD_4        NOP /// P=&T[5], add_two(p[1], *(p+2)); ////////////////////////
                           125
4101             0000:0159 126              LD              A B
1605 0002 0000   0000:0160 127              SUB             A #2
1605 0010 0000   0000:0163 128              SUB             A #(5*2)
5205 9972        0000:0166 129              ST              A %-28          // PTR_P = #(TABLE[5])
                           130
4245 9972        0000:0168 131              LD              X %-28          // GET *PTR_P
1645 0002 0000   0000:0170 132              SUB             X #(1*2)        // = PTR_P[1]
4302 0000 0000   0000:0173 133              LD              A @0
9880             0000:0176 134              PUSH            A
4245 9972        0000:0177 135              LD              X %-28
4302 9996 9999   0000:0179 136              LD              A @(-2*2)       // = *(PTR_P+2)
9880             0000:0182 137              PUSH            A               // PARAMETER
9881             0000:0183 138              PUSH            B                       // B->DATA BACKUP
```

```
                                 139
6700 0007 0000   0000:0184 140              CALL            ADD_TWO
                                 141
4235 9998        0000:0187 142              LD              D %-2           // GET RETURN_VALUE(TO D)
9891             0000:0189 143              POP             B
5235 9974        0000:0190 144              ST              D %-26          // RLST = RETURN_VALIE
1355 0004 0000   0000:0192 145              ADD             SP #(2*2)       // RESET ST
                                 146
4205 9974        0000:0195 147              LD              A %-26
0612             0000:0197 148              OUT             12 ///////// OUT(RSLT) ////////////////////////
                                 149
4305 0010 0000   0000:0198 150              LD              A #10
0613             0000:0201 151              OUT             SCREEN_TXT
                                 152
4151             0000:0202 153              LD              SP B
9999             0000:0203 154              RET
                                 155
                                 156
                 0000:0204 157 HEAP                    RESDBOX 100
                                 158 STK_BTM                 ORG     $
                                 159 END
```

[다음은 실행결과입니다.]

```
% Successfully Loaded, Type "RUN 0"
Monitor> run 0
% RUN 00000000
0015
0011
0017
0007

Shutdown Little Man Computer!

SCORE : 29/80 (not impl.)/(total)
1. INSTRUCTION SCORE : 29/80(s) not implemented instructions
2. MAGICCODE SCORE : called 1(s)/25(s) kinds
[u20103390@linux LMC-1.3.4.6]$
```


[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
[9]
. . .
TMP
RET_VAL
&MAIN
B(FIRST)
D(PARAM)
A(PARAM)
I
PTR_P
RSLT
B
A
TABLE 9
TABLE 8
TABLE 7
TABLE 6
TABLE 5
TABLE 4
TABLE 3
TABLE 2
TABLE 1
TABLE 0
&FIRST
%-28
%-22
%-2
%-2
%-26
B
SP