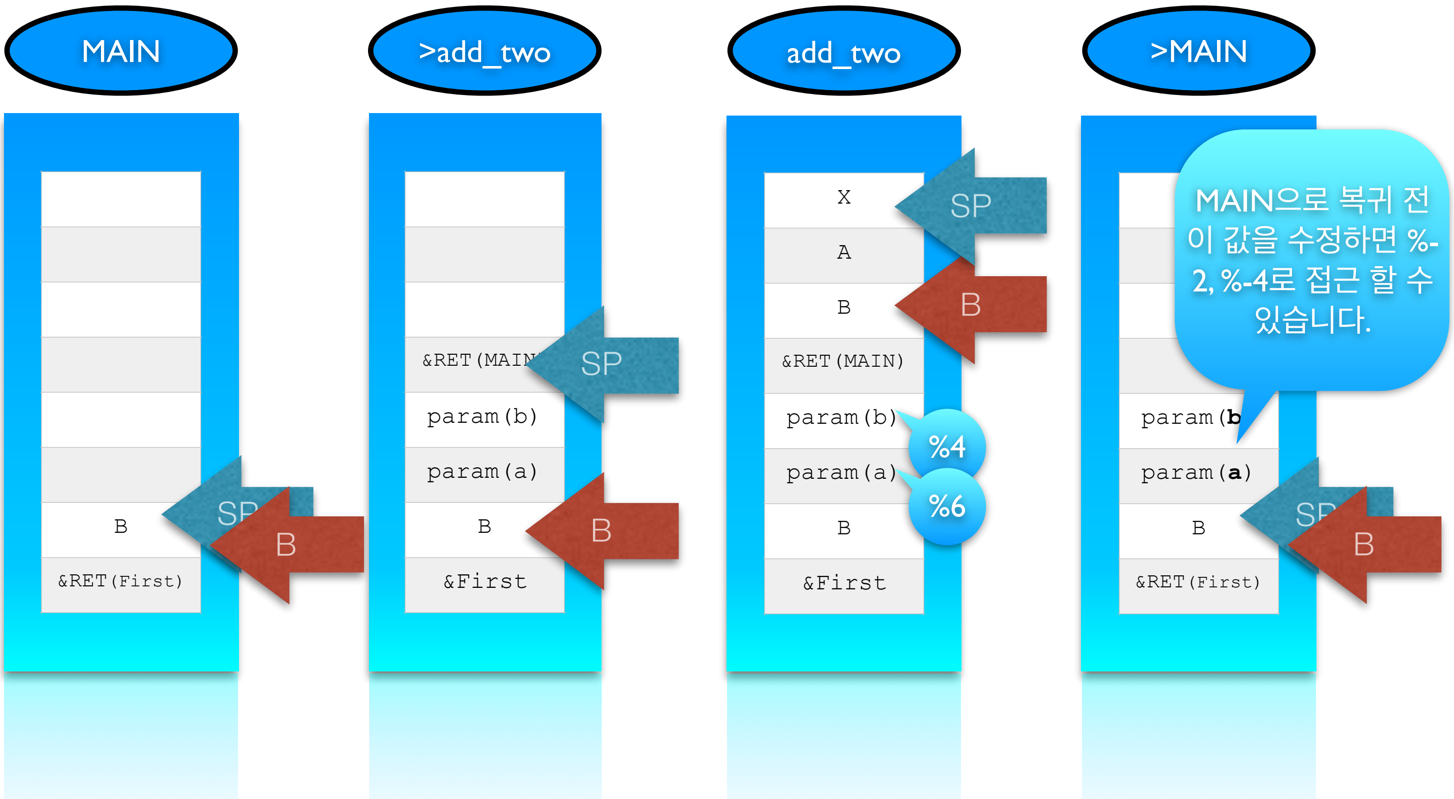
1. 다음 프로그램의 각 변수들 배치상황을 설명하시오.

```
static int table[10] = {10,9,8,7,6,5,4,3,2,1};
static int a, b, result;
static int *p;

static int add_two(int x, int y) {
    int tmp;
    tmp = x + y;
    return tmp;
    }

main() {

    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);

}
```

Static으로 선언하므로써, 스코프를 이 모듈 내에서만 으로 한정하게 하였습니다.

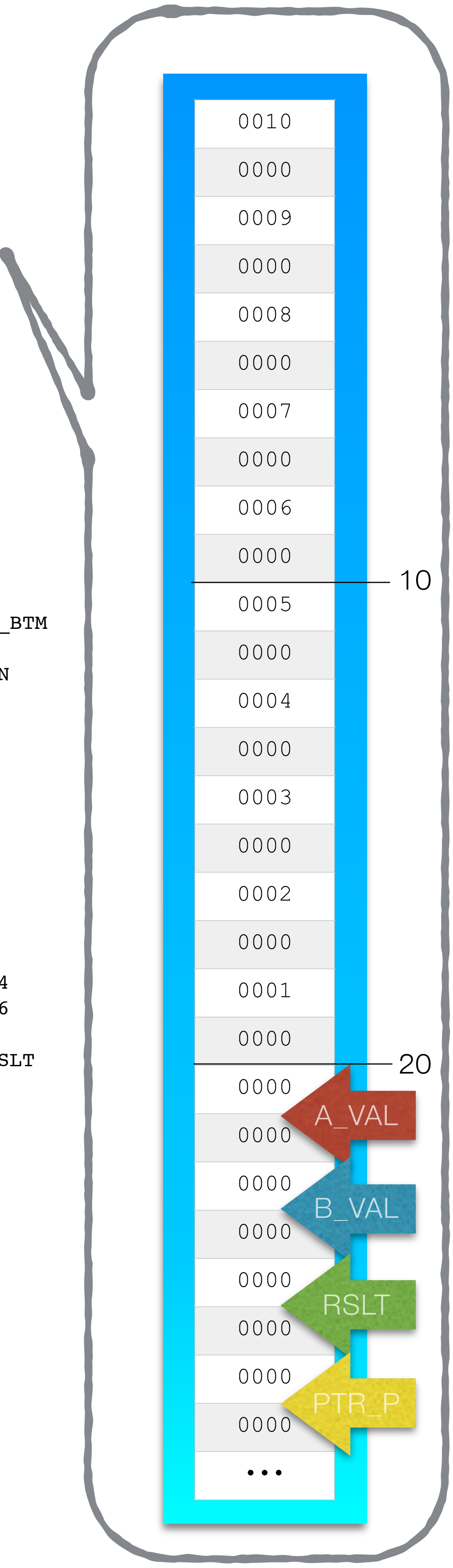
파라미터로 넘어온 x, y와 tmp변수는 함수가 끝나면 사라집니다.

main함수에선 Static으로 선언된 변수들을 사용합니다. ADD_TWO 함수로 변수를 넘겨주기 위해 Activation Record를 사용합니다. 이는 어셈블리어의 PUSH명령을 통해 구현하며, 함수로 넘어와 사용할 때, B Register를 통해 스텍 영역에 있는 파라미터 변수에 접근할 수 있습니다. 따라서 파라미터로 넘겨 줄 변수에 따로 라벨을 달아줄 필요가 없습니다.



2. 프로그램을 LMC코드로 바꾸어 모니터를 통해 나온 어셈블리 리스트를 보이시오.

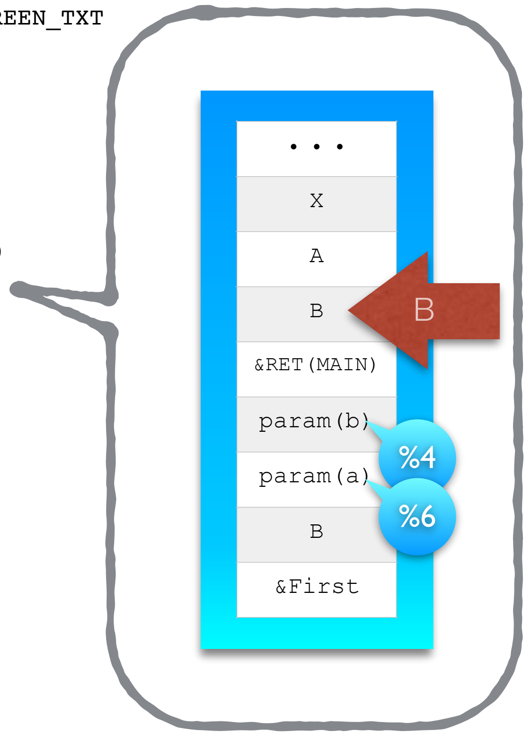
```
20103390> asm 4 5 6
Execute% Assemble 4:cassette/simpleProg.lmc
                 0000:0000   1 SIMPLE_PROG     START   0
                 0000:0000   2 KEYBOARD_DEC    EQU     10
                 0000:0000   3 SCREEN_DEC      EQU     12
                 0000:0000   4 SCREEN_TXT      EQU     13
                             5
0010 0000        0000:0000   6 TABLE           DBOX    10
0009 0000        0000:0002   7                 DBOX    9
0008 0000        0000:0004   8                 DBOX    8
0007 0000        0000:0006   9                 DBOX    7
0006 0000        0000:0008  10                 DBOX    6
0005 0000        0000:0010  11                 DBOX    5
0004 0000        0000:0012  12                 DBOX    4
0003 0000        0000:0014  13                 DBOX    3
0002 0000        0000:0016  14                 DBOX    2
0001 0000        0000:0018  15                 DBOX    1
                 0000:0020  16 A_VAL           RESDBOX 1
                 0000:0022  17 B_VAL           RESDBOX 1
                 0000:0024  18 RSLT            RESDBOX 1
                 0000:0026  19 PTR_P           RESDBOX 1
                            20
                            21 //#begin
                            21 //% MOV SP #STK_BTM
4355 0425 0000   0000:0028  21                 LD   SP #STK_BTM
                            21 //#end
6700 0052 0000   0000:0031  22                 CALL    MAIN
0700             0000:0034  23                 COB
                            24
9881             0000:0035  25 ADD_TWO         PUSH    B
                            26 //#begin
                            26 //% MOV B SP
4115             0000:0036  26                 LD   B  SP
                            26 //#end
                            27
9880             0000:0037  28                 PUSH    A
9884             0000:0038  29                 PUSH    X
                            30
4205 0004        0000:0039  31                 LD      A %4
4245 0006        0000:0041  32                 LD      X %6
1104             0000:0043  33                 ADD     A X
5300 0024 0000   0000:0044  34                 ST      A RSLT
                            35
9894             0000:0047  36                 POP     X
9890             0000:0048  37                 POP     A
                            38
                            39 //#begin
                            39 //% MOV SP B
4151             0000:0049  39                 LD   SP B
                            39 //#end
9891             0000:0050  40                 POP     B
9999             0000:0051  41                 RET
                            42
                            43
                 0000:0052  44 MAIN            RESBOX  1
                            45 //#begin
                            45 //% MOV B SP
4115             0000:0053  45                 LD   B  SP
```



```
                            45 //#end
                            46
                            47 //#begin
                            47 //% MOV A #2
4305 0002 0000   0000:0054  47                LD  A #2
                            47 //#end
9880             0000:0057  48                PUSH    A
                            49 //#begin
                            49 //% MOV A #3
4305 0003 0000   0000:0058  49                LD  A #3
                            49 //#end
9880             0000:0061  50                PUSH    A
6700 0035 0000   0000:0062  51                CALL    ADD_TWO
                            52 //#begin
                            52 //% MOV SP B
4151             0000:0065  52                LD  SP B
                            52 //#end
                            53
4300 0024 0000   0000:0066  54                LD      A RSLT
0612             0000:0069  55                OUT     SCREEN_DEC
                            56 //#begin
                            56 //% MOV A #10
4305 0010 0000   0000:0070  56                LD  A #10
                            56 //#end
0613             0000:0073  57                OUT     SCREEN_TXT
                            58
                            59 //#begin
                            59 //% MOV     B SP
4115             0000:0074  59                LD  B SP
                            59 //#end
                            60 //#begin
                            60 //% MOV     A TABLE+(2*2)
4300 0004 0000   0000:0075  60                LD  A 4
                            60 //#end
5300 0020 0000   0000:0078  61                ST      A A_VAL
                            62
                            63 //#begin
                            63 //% MOV     A TABLE+(3*2)
4300 0006 0000   0000:0081  63                LD  A 6
                            63 //#end
5300 0022 0000   0000:0084  64                ST      A B_VAL
                            65
4300 0020 0000   0000:0087  66                LD      A A_VAL
9880             0000:0090  67                PUSH    A
4300 0022 0000   0000:0091  68                LD      A B_VAL
9880             0000:0094  69                PUSH    A
6700 0035 0000   0000:0095  70                CALL    ADD_TWO
                            71 //#begin
                            71 //% MOV     SP B
4151             0000:0098  71                LD  SP B
                            71 //#end
                            72
4300 0024 0000   0000:0099  73                LD      A RSLT
0612             0000:0102  74                OUT     SCREEN_DEC
                            75 //#begin
                            75 //% MOV     A #10
4305 0010 0000   0000:0103  75                LD  A #10
                            75 //#end
0613             0000:0106  76                OUT     SCREEN_TXT
                            77
                            78 //#begin
                            78 //% MOV B SP
4115             0000:0107  78                LD  B SP
                            78 //#end
                            79 //#begin
                            79 //% MOV A TABLE+(4*2)
```

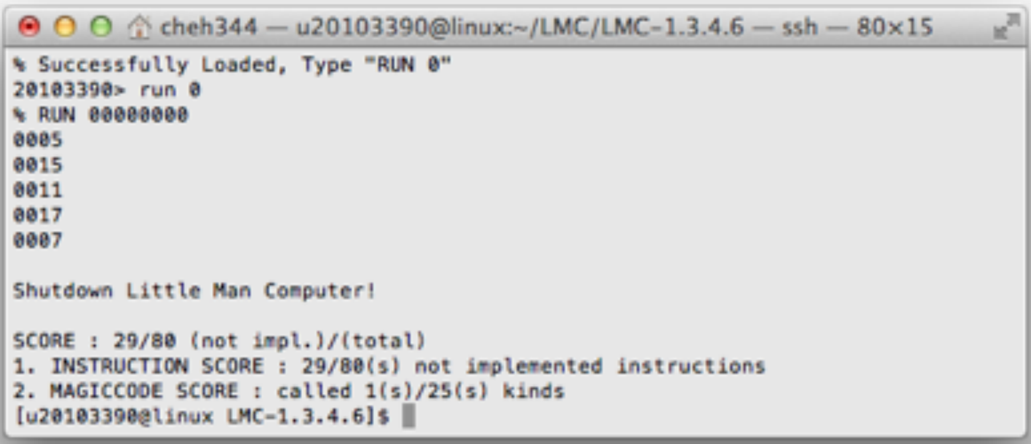
```
4300 0008 0000   0000:0108  79                 LD  A 8
                            79 //#end
9880             0000:0111  80                 PUSH    A
                            81 //#begin
                            81 //% MOV A TABLE+(5*2)
4300 0010 0000   0000:0112  81                 LD  A 10
                            81 //#end
9880             0000:0115  82                 PUSH    A
6700 0035 0000   0000:0116  83                 CALL    ADD_TWO
                            84 //#begin
                            84 //% MOV SP B
4151             0000:0119  84                 LD  SP B
                            84 //#end
                            85
4300 0024 0000   0000:0120  86                 LD      A RSLT
0612             0000:0123  87                 OUT     SCREEN_DEC
                            88 //#begin
                            88 //% MOV A #10
4305 0010 0000   0000:0124  88                 LD  A #10
                            88 //#end
0613             0000:0127  89                 OUT     SCREEN_TXT
                            90
4305 0000 0000   0000:0128  91                 LD      A #TABLE
5300 0026 0000   0000:0131  92                 ST      A PTR_P
                            93
                            94 //#begin
                            94 //% MOV B SP
4115             0000:0134  94                 LD  B SP
                            94 //#end
4300 0026 0000   0000:0135  95                 LD      A PTR_P
1305 0002 0000   0000:0138  96                 ADD     A #(1*2)
5300 0020 0000   0000:0141  97                 ST      A A_VAL
4301 0020 0000   0000:0144  98                 LD      A *A_VAL
9880             0000:0147  99                 PUSH    A
                           100
4300 0026 0000   0000:0148 101                 LD      A PTR_P
5300 0022 0000   0000:0151 102                 ST      A B_VAL
1305 0004 0000   0000:0154 103                 ADD     A #(2*2)
5300 0022 0000   0000:0157 104                 ST      A B_VAL
4301 0022 0000   0000:0160 105                 LD      A *B_VAL
9880             0000:0163 106                 PUSH    A
6700 0035 0000   0000:0164 107                 CALL    ADD_TWO
                           108
4300 0024 0000   0000:0167 109                 LD      A RSLT
0612             0000:0170 110                 OUT     SCREEN_DEC
                           111 //#begin
                           111 //% MOV A #10
4305 0010 0000   0000:0171 111                 LD  A #10
                           111 //#end
0613             0000:0174 112                 OUT     SCREEN_TXT
                           113
                           114 //#begin
                           114 //% MOV SP B
4151             0000:0175 114                 LD  SP B
                           114 //#end
                           115
4305 0010 0000   0000:0176 116                 LD      A #TABLE+5*2
5300 0026 0000   0000:0179 117                 ST      A PTR_P
                           118
                           119 //#begin
                           119 //% MOV     B SP
4115             0000:0182 119                 LD  B SP
                           119 //#end
4300 0026 0000   0000:0183 120                 LD      A PTR_P
1305 0002 0000   0000:0186 121                 ADD     A #(1*2)
5300 0020 0000   0000:0189 122                 ST      A A_VAL
```

```
4301 0020 0000    0000:0192 123                LD      A *A_VAL
9880              0000:0195 124                PUSH    A
                            125
4300 0026 0000    0000:0196 126                LD      A PTR_P
5300 0022 0000    0000:0199 127                ST      A B_VAL
1305 0004 0000    0000:0202 128                ADD     A #(2*2)
5300 0022 0000    0000:0205 129                ST      A B_VAL
4301 0022 0000    0000:0208 130                LD      A *B_VAL
9880              0000:0211 131                PUSH    A
6700 0035 0000    0000:0212 132                CALL    ADD_TWO
                            133
4300 0024 0000    0000:0215 134                LD      A RSLT
0612              0000:0218 135                OUT     SCREEN_DEC
                            136 //#begin
                            136 //% MOV A #10
4305 0010 0000    0000:0219 136                LD  A #10
                            136 //#end
0613              0000:0222 137                OUT     SCREEN_TXT
                            138
                            139 //#begin
                            139 //% MOV SP B
4151              0000:0223 139                LD  SP B
                            139 //#end
9999              0000:0224 140                RET
                            141
                  0000:0225 142                RESDBOX 100
                  0000:0425 143 STK_BTM        EQU     $
                            144
                            145
                            146 END
```



* **다음은 실행예시입니다.**

- **3번 카세트 : NEW_BOOT**
  4번 카세트 : simpleProg.lmc
  5번 카세트 : simpleProg.bl
  6번 카세트 : silpleProg.list
  가 장착되어 있습니다.



```
% Successfully Loaded, Type "RUN 0"
20103390> run 0
% RUN 00000000
0005
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]$
```