

Compiler

과제8. Code generation

20201696 안선영

1. 개요

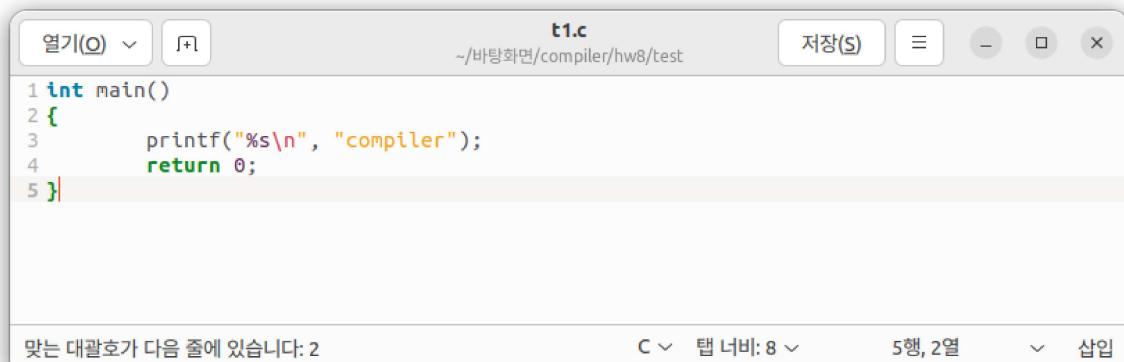
그동안 만들었던 컴파일러 프로그램에서 코드 생성기 프로그램을 추가한 뒤 최종 C 컴파일러를 완성하였습니다. **교재의 A8 code generator**코드와 **A9의 main()**함수를 참고하여 코드를 작성하였습니다. 테스트 파일은 기본 printf, scanf, malloc, sort를 테스트할 수 있는 파일을 만들어 실험하였습니다.

결과가 정상적으로 출력 되는지 점검하기 위해 syntax analysis와 semantic analysis를 txt파일로 저장한 뒤 작성한 테스트 프로그램의 문법 및 구문 오류가 있을 경우, 이 파일을 참고하여 오류 코드를 수정하였습니다.

보고서는 1. 개요 2. 실행 결과 3 . 실행 결과에 대한 syntax 및 semantic tree 출력 4. 소스 파일 및 디렉토리 구조 참고 사항 순서로 작성하였습니다.

2. 실행 결과 (7개 테스트)

1) 'compiler'라는 문자열을 출력하는 프로그램



```
t1.c
~/바탕화면/compiler/hw8/test
1 int main()
2 {
3     printf("%s\n", "compiler");
4     return 0;
5 }|
```

맞는 대괄호가 다음 줄에 있습니다: 2

C ↕ 탭 너비: 8 ↕ 5행, 2열 ↕ 삽입

```
asy@asy-H410M-DS2V:~/바탕화면/compiler/hw8/test$ ./a.out < t1.c > t1.txt
asy@asy-H410M-DS2V:~/바탕화면/compiler/hw8/test$ ./i.out a.asm
===== symbol =====
1: printf -1
2: malloc -2
3: scanf -3
4: main 3
===== code =====
0: INT 0,32
1: SUP 0,3
2: RET 0,0
3: INT 0,12
4: INT 0,12
5: LDA 0,12
6: LDA 0,20
7: POP 0,5
8: ADDR 0,-1
9: CAL 0,0
10: LDA 1,-4
11: LITI 0,0
12: STO 0,1
13: RET 0,0
14: RET 0,0
start execution
compiler
end execution
asy@asy-H410M-DS2V:~/바탕화면/compiler/hw8/test$
```

2) 0~4까지 출력 되는 반복문 프로그램

```
t2.c
~/바탕화면/.compiler/hw8/test
1 int main()
2 {
3     int i;
4     for (i = 0; i < 5; ++i) {
5         printf("%d\n", i);
6     }
7     return 0;
8 }
```

C 템 너비: 8 8행, 2열 삽입

```
asy@asy-H410M-DS2V: ~/바탕화면/.compiler/hw8/test$ ./a.out < t2.c > t2.txt
asy@asy-H410M-DS2V: ~/바탕화면/.compiler/hw8/test$ ./i.out a.asm
===== symbol =====
1: printf -1
2: malloc -2
3: scanf -3
4: main 3
5: L2 8
6: L3 24
7: L1 18
===== code =====
0: INT 0,20
1: SUP 0,3
2: RET 0,0
3: INT 0,16
4: LDA 1,12
5: LITI 0,0
6: STX 0,1
7: POP 0,1
8: LOD 1,12
9: LITI 0,5
10: LSSI 0,0
11: JPC 0,24
12: INT 0,12
13: LDA 0,12
14: LOD 1,12
15: POP 0,5
16: ADDR 0,-1
17: CAL 0,0
18: LDA 1,12
19: LDX 0,1
20: INCI 0,0
21: STX 0,1
22: POP 0,1
23: JMP 0,8
24: LDA 1,-4
```

```
asy@asy-H410M-DS2V: ~/바탕화면/compiler/hw8/test
```

```
25: LITI    0,0
26: STO     0,1
27: RET     0,0
28: RET     0,0
start execution
0
1
2
3
4
end execution
asy@asy-H410M-DS2V:~/바탕화면/.compiler/hw8/test$
```

3) 덧셈 프로그램

```
열기(O) ▾  [+]
t3.c
~/바탕화면/.compiler/hw8/test
저장(S)  ⋮  -  □  ×
```

```
1 int main()
2 {
3     int a;
4     int b;
5     int sum;
6     a = 100;
7     b = 200;
8     sum = a + b;
9     printf("%d\n", sum);
10    return 0;
11 }
```

```
C ⓘ  탭 너비: 8 ⓘ  9행, 29열  ⓘ  삽입
```

```
asy@asy-H410M-DS2V: ~/바탕화면/compiler/hw8/test$ ./a.out < t3.c > t3.txt
asy@asy-H410M-DS2V:~/바탕화면/compiler/hw8/test$ ./i.out a.asm
===== symbol =====
1: printf -1
2: malloc -2
3: scanf -3
4: main 3
===== code =====
0: INT 0,20
1: SUP 0,3
2: RET 0,0
3: INT 0,24
4: LDA 1,12
5: LITI 0,100
6: STX 0,1
7: POP 0,1
8: LDA 1,16
9: LITI 0,200
10: STX 0,1
11: POP 0,1
12: LDA 1,20
```

```
asy@asy-H410M-DS2V: ~/바탕화면/compiler/hw8/test$ ./i.out a.asm
13: LOD 1,12
14: LOD 1,16
15: ADDI 0,0
16: STX 0,1
17: POP 0,1
18: INT 0,12
19: LDA 0,12
20: LOD 1,20
21: POP 0,5
22: ADDR 0,-1
23: CAL 0,0
24: LDA 1,-4
25: LITI 0,0
26: STO 0,1
27: RET 0,0
28: RET 0,0
start execution
300
end execution
asy@asy-H410M-DS2V:~/바탕화면/compiler/hw8/test$
```

4) scanf로 입력 받은 x,y값을 바꾸는 프로그램

```
t3.c
1 int main() {
2     int x, y, temp;
3     scanf("%d %d", &x, &y);
4
5     temp = x;
6     x = y;
7     y = temp;
8
9     printf("%d %d\n", x, y);
10
11    return 0;
12 }
```

```
t4.c
~/바탕화면/.compiler/hw8/test
```

C 템 너비: 8 7행, 18열 삽입

```
asy@asy-H410M-DS2V:~/바탕화면/.compiler/hw8/test$ ./a.out < t4.c > t4.txt
asy@asy-H410M-DS2V:~/바탕화면/.compiler/hw8/test$ ./i.out a.asm
===== symbol =====
1: printf -1
2: malloc -2
3: scanf -3
4: main 3
===== code =====
0: INT 0,32
1: SUP 0,3
2: RET 0,0
3: INT 0,24
4: INT 0,12
5: LDA 0,12
6: LDA 1,12
7: LDA 1,16
8: POP 0,6
9: ADDR 0,-3
10: CAL 0,0
11: LDA 1,20
12: LOD 1,12
13: STX 0,1
14: POP 0,1
15: LDA 1,12
```

```
asy@asy-H410M-DS2V: ~/바탕화면/.compiler/hw8/test
16: LOD      1,16
17: STX      0,1
18: POP      0,1
19: LDA      1,16
20: LOD      1,20
21: STX      0,1
22: POP      0,1
23: INT      0,12
24: LDA      0,20
25: LOD      1,12
26: LOD      1,16
27: POP      0,6
28: ADDR     0,-1
29: CAL      0,0
30: LDA      1,-4
31: LITI     0,0
32: STO      0,1
33: RET      0,0
34: RET      0,0
start execution
100 200
200 100
end execution
asy@asy-H410M-DS2V:~/바탕화면/.compiler/hw8/test$
```

5) malloc한 뒤 배열의 값을 출력하는 프로그램

```
t5.c
~/바탕화면/.compiler/hw8/test
t3.c           t4.c           t5.c
열기(O)  ▾  ↳
저장(S)  ⍎  ⍎  ⍎
1 void main()
2 {
3     int *a;
4     int count;
5     int i;
6     count = 10;
7     a = malloc(sizeof(int) *count);
8     for (i = 0; i < count; ++i)
9     {
10         a[i] = i;
11     }
12     for (i = 0; i < count; ++i)
13     {
14         printf("number[%d] : %d\n", i + 1, *(a + i));
15     }
16 }
```

```
asy@asy-H410M-DS2V: ~/바탕화면/.compiler/hw8/test$ ./a.out < t5.c > t5.txt
asy@asy-H410M-DS2V: ~/바탕화면/.compiler/hw8/test$ ./i.out a.asm
===== symbol =====
1: printf -1
2: malloc -2
3: scanf -3
4: main 3
5: L2 22
6: L3 40
7: L1 34
8: L5 44
9: L6 68
10: L4 62
===== code =====
0: INT 0,32
1: SUP 0,3
2: RET 0,0
3: INT 0,24
4: LDA 1,16
5: LITI 0,10
6: STX 0,1
7: POP 0,1
8: LDA 1,12
9: INT 0,16
10: LITI 0,4
11: LOD 1,16
12: MULI 0,0
13: POP 0,4
14: ADDR 0.-2
```

```
asy@asy-H410M-DS2V: ~/바탕화면/.compiler/hw8/test
15: CAL 0,0
16: STX 0,1
17: POP 0,1
18: LDA 1,20
19: LITI 0,0
20: STX 0,1
21: POP 0,1
22: LOD 1,20
23: LOD 1,16
24: LSSI 0,0
25: JPC 0,40
26: LOD 1,12
27: LOD 1,20
28: LITI 0,4
29: MULI 0,0
30: OFFSET 0,0
31: LOD 1,20
32: STX 0,1
33: POP 0,1
34: LDA 1,20
35: LDX 0,1
36: INCI 0,0
37: STX 0,1
38: POP 0,1
39: JMP 0,22
40: LDA 1,20
41: LITI 0,0
42: STX 0,1
43: POP 0,1
```

```
asy@asy-H410M-DS2V: ~/바탕화면/compiler/hw8/test
```

```
44: LOD      1,20
45: LOD      1,16
46: LSSI     0,0
47: JPC      0,68
48: INT      0,12
49: LDA      0,12
50: LOD      1,20
51: LITI     0,1
52: ADDI     0,0
53: LOD      1,12
54: LOD      1,20
55: LITI     0,4
56: MULI     0,0
57: ADDI     0,0
58: LDI      0,1
59: POP      0,6
60: ADDR     0,-1
61: CAL      0,0
62: LDA      1,20
63: LDX      0,1
64: INCI     0,0
65: STX      0,1
66: POP      0,1
67: JMP      0,44
68: RET      0,0

start execution
number[1] : 0
number[2] : 1
number[3] : 2
number[4] : 3
number[5] : 4
number[6] : 5
number[7] : 6
number[8] : 7
number[9] : 8
number[10] : 9
end execution
```

6) scanf로 입력 받은 두 값을 사칙연산 등 여러가지 결과를 출력하는 프로그램

A screenshot of a terminal window titled "t6.c" showing the source code for a C program. The code defines a main function that reads two integers from the user, performs various arithmetic operations (addition, subtraction, multiplication, division, modulus), and prints the results. The code is as follows:

```
1 int main()
2 {
3     int a;
4     int b;
5
6     scanf("%d %d", &a, &b);
7
8     printf("result : %d\n", a + b);
9     printf("result : %d\n", a - b);
10    printf("result : %d\n", a * b);
11    printf("result : %d\n", a / b);
12    printf("result : %d\n", a % b);
13
14    return 0;
15 }
```

The terminal window also shows the file path: ~/바탕화면/.compiler/hw8/test. At the bottom, it displays "C 템 네비: 8" and "6행, 32열".

A screenshot of a terminal window showing the assembly code generated by the compiler. The command run was "./i.out a.asm". The output shows symbols and assembly instructions. The symbols are:

```
===== symbol =====
1: printf -1
2: malloc -2
3: scanf -3
4: main 3
```

The assembly code is:

```
===== code =====
0: INT 0,100
1: SUP 0,3
2: RET 0,0
3: INT 0,20
4: INT 0,12
5: LDA 0,12
6: LDA 1,12
7: LDA 1,16
8: POP 0,6
9: ADDR 0,-3
10: CAL 0,0
11: INT 0,12
12: LDA 0,20
13: LOD 1,12
14: LOD 1,16
15: ADDI 0,0
16: POP 0,5
17: ADDR 0,-1
18: CAL 0,0
19: INT 0,12
20: LDA 0,36
21: LOD 1,12
22: LOD 1,16
23: SUBI 0,0
24: POP 0,5
25: ADDR 0,-1
26: CAL 0,0
27: INT 0,12
28: LDA 0,52
```

```
asy@asy-H410M-DS2V: ~/바탕화면/.compiler/hw8/test
29: LOD      1,12
30: LOD      1,16
31: MULI    0,0
32: POP      0,5
33: ADDR    0,-1
34: CAL      0,0
35: INT      0,12
36: LDA      0,68
37: LOD      1,12
38: LOD      1,16
39: DIVI    0,0
40: POP      0,5
41: ADDR    0,-1
42: CAL      0,0
43: INT      0,12
44: LDA      0,84
45: LOD      1,12
46: LOD      1,16
47: MOD      0,0
48: POP      0,5
49: ADDR    0,-1
50: CAL      0,0
51: LDA      1,-4
52: LITI    0,0
53: STO      0,1
54: RET      0,0
55: RET      0,0
start execution
3 10
result : 13
result : -7
result : 30
result : 0
result : 3
end execution
asy@asy-H410M-DS2V:~/바탕화면/.compiler/hw8/test$
```

7) sort 프로그램

```
열기(O) ▾ ⌂ t7.c ~/바탕화면/compiler/hw8/test 저장(S) ⌂ - □ × t3.c × t4.c × t5.c × t6.c × t7.c ×
1 int a[5];
2 void sort(int a[ ], int n) {
3     int x, y, z;
4     for (x=0; x<n; x++)
5         for (y=x+1; y<n; y=y+1)
6             if (a[x]<a[y]) {
7                 z=a[x]; a[x]=a[y]; a[y]=z;}
8 }
9
10
11 int main() {
12     int i;
13     a[0] = 3;
14     a[1] = 1;
15     a[2] = 4;
16     a[3] = 2;
17     a[4] = 5;
18     sort(a, 5);
19     for (i = 0; i < 5; i++) {
20         printf("%d ", a[i]);
21     }
22     printf("\n");
23 }
```

```
asy@asy-H410M-DS2V: ~/바탕화면/compiler/hw8/test$ ./a.out < t8.c > t8.txt
asy@asy-H410M-DS2V: ~/바탕화면/compiler/hw8/test$ ./i.out a.asm
===== symbol =====
1: printf -1
2: malloc -2
3: scanf -3
4: main 81
5: sort 3
6: L2 8
7: L3 80
8: L5 18
9: L6 73
10: L7 66
11: L4 66
12: L1 73
13: L9 132
14: L10 154
15: L8 147
===== code =====
0: INT 0,48
1: SUP 0,81
2: RET 0,0
3: INT 0,32
4: LDA 1,20
5: LITI 0,0
6: STX 0,1
7: POP 0,1
8: LOD 1,20
9: LOD 1,16
10: LSSI 0,0
11: JPC 0,80
12: LDA 1,24
13: LOD 1,20
14: LITI 0,1
15: ADDI 0,0
16: STX 0,1
17: POP 0,1
```

```
asy@asy-H410M-DS2V: ~/바탕화면/compiler/hw8/test
```

```
18: LOD      1,24
19: LOD      1,16
20: LSSI     0,0
21: JPC      0,73
22: LOD      1,12
23: LOD      1,20
24: LITI     0,4
25: MULI     0,0
26: OFFSET   0,0
27: LDI      0,1
28: LOD      1,12
29: LOD      1,24
30: LITI     0,4
31: MULI     0,0
32: OFFSET   0,0
33: LDI      0,1
34: LSSI     0,0
35: JPC      0,66
36: LDA      1,28
37: LOD      1,12
38: LOD      1,20
39: LITI     0,4
40: MULI     0,0
41: OFFSET   0,0
42: LDI      0,1
43: STX      0,1
44: POP      0,1
45: LOD      1,12
46: LOD      1,20
47: LITI     0,4
48: MULI     0,0
49: OFFSET   0,0
50: LOD      1,12
51: LOD      1,24
52: LITI     0,4
53: MULI     0,0
```

```
asy@asy-H410M-DS2V: ~/바탕화면/compiler/hw8/test
```

```
54:  OFFSET  0,0
55:  LDI      0,1
56:  STX      0,1
57:  POP     0,1
58:  LOD      1,12
59:  LOD      1,24
60:  LITI     0,4
61:  MULI     0,0
62:  OFFSET   0,0
63:  LOD      1,28
64:  STX      0,1
65:  POP     0,1
66:  LDA      1,24
67:  LOD      1,24
68:  LITI     0,1
69:  ADDI     0,0
70:  STX      0,1
71:  POP     0,1
72:  JMP      0,18
73:  LOD      1,20
74:  LDA      1,20
75:  LDX      0,1
76:  INCI     0,0
77:  STO      0,1
78:  POP     0,1
79:  JMP      0,8
80:  RET      0,0
81:  INT      0,16
82:  LOD      1,12
83:  LITI     0,0
84:  LITI     0,4
85:  MULI     0,0
86:  OFFSET   0,0
87:  LITI     0,3
88:  STX      0,1
89:  POP     0,1
```



asy@asy-H410M-DS2V: ~/바탕화면/.compiler/hw8/test



```
89: POP      0,1
90: LOD      1,12
91: LITI     0,1
92: LITI     0,4
93: MULI    0,0
94: OFFSET   0,0
95: LITI     0,1
96: STX      0,1
97: POP      0,1
98: LOD      1,12
99: LITI     0,2
100: LITI    0,4
101: MULI   0,0
102: OFFSET  0,0
103: LITI    0,4
104: STX     0,1
105: POP     0,1
106: LOD     1,12
107: LITI    0,3
108: LITI    0,4
109: MULI   0,0
110: OFFSET  0,0
111: LITI    0,2
112: STX     0,1
113: POP     0,1
114: LOD     1,12
115: LITI    0,4
116: LITI    0,4
117: MULI   0,0
118: OFFSET  0,0
119: LITI    0,5
120: STX     0,1
121: POP     0,1
122: INT     0,12
123: LOD     1,12
124: LITI    0,5
```

```
asy@asy-H410M-DS2V: ~/바탕화면/compiler/hw8/test
125: POP      0,5
126: ADDR    0,3
127: CAL      0,0
128: LDA      1,12
129: LITI     0,0
130: STX      0,1
131: POP      0,1
132: LOD      1,12
133: LITI     0,5
134: LSSI     0,0
135: JPC      0,154
136: INT      0,12
137: LDA      0,32
138: LOD      1,12
139: LOD      1,12
140: LITI     0,4
141: MULI     0,0
142: OFFSET   0,0
143: LDI      0,1
144: POP      0,5
145: ADDR    0,-1
146: CAL      0,0
147: LOD      1,12
148: LDA      1,12
149: LDX      0,1
150: INCI     0,0
151: STO      0,1
152: POP      0,1
153: JMP      0,132
154: INT      0,12
155: LDA      0,40
156: POP      0,4
157: ADDR    0,-1
158: CAL      0,0
159: RET      0,0
start execution
```

```

asy@asy-H410M-DS2V: ~/바탕화면/compiler/hw8/test
155: LDA      0,40
156: POP     0,4
157: ADDR    0,-1
158: CAL      0,0
159: RET      0,0
start execution
5 4 3 2 0
end execution
asy@asy-H410M-DS2V:~/바탕화면/compiler/hw8/test$ 

```

3. 실행 결과에 대한 syntax 및 semantic tree 출력

1) 'compiler'라는 문자열을 출력하는 프로그램

```

===== syntax tree =====
N_PROGRAM (0,0)
| (ID="main") TYPE:72241b60 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
|| TYPE
||| FUNCTION
||| PARAMETER
||| TYPE
||| (int)
||| BODY
|||| N_STMT_COMPOUND (0,0)
||||| N_STMT_LIST (0,0)
||||| N_STMT_EXPRESSION (0,0)
||||| N_EXP_FUNCTION_CALL (0,0)
||||| N_EXP_IDENT (0,0)
||||| (ID="printf") TYPE:7223c7a0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||||| N_ARG_LIST (0,0)
||||| N_EXP_STRING_LITERAL (0,0)
||||| "%s\n"
||||| N_ARG_LIST (0,0)
||||| N_EXP_STRING_LITERAL (0,0)
||||| "compiler"
||||| N_ARG_LIST_NIL (0,0)
||||| N_STMT_LIST (0,0)
||||| N_STMT_RETURN (0,0)
||||| N_EXP_INT_CONST (0,0)
||||| 0
||||| N_STMT_LIST_NIL (0,0)
===== semantic tree =====
N_PROGRAM (0,32)
| (ID="main") TYPE:72241b60 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
|| TYPE

```

```

||| FUNCTION
|||| PARAMETER
|||| TYPE
||||| (int)
|||| BODY
||||| N_STMT_COMPOUND (0,0)
||||| N_STMT_LIST (0,0)
|||||| N_STMT_EXPRESSION (0,0)
|||||| N_EXP_FUNCTION_CALL (7223c680,0)
|||||| N_EXP_AMP (72242fb0,0)
|||||| N_EXP_IDENT (7223c7a0,0)
|||||||| (ID="printf") TYPE:7223c7a0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||||||| N_ARG_LIST (0,8)
||||||| N_EXP_STRING_LITERAL (7223c6c0,0)
||||||| LITERAL: "%s\n"
||||||| N_ARG_LIST (0,4)
||||||| N_EXP_STRING_LITERAL (7223c6c0,0)
||||||| LITERAL: "compiler"
||||||| N_ARG_LIST_NIL (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_RETURN (0,0)
||||||| N_EXP_INT_CONST (7223c4d0,0)
||||||| INT=0
||||||| N_STMT_LIST_NIL (0,0)

```

2) 0~4까지 출력 되는 반복문 프로그램

```

===== syntax tree =====
N_PROGRAM (0,0)
| (ID="main") TYPE:e4a1db60 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
|| TYPE
||| FUNCTION
|||| PARAMETER
|||| TYPE
||||| (int)
|||| BODY
||||| N_STMT_COMPOUND (0,0)
||||| (ID="i") TYPE:e4a184d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||| TYPE
||||| (int)
||||| N_STMT_LIST (0,0)
||||| N_STMT_FOR (0,0)
||||| N_FOR_EXP (0,0)
||||| N_EXP_ASSIGN (0,0)
||||| N_EXP_IDENT (0,0)
||||| (ID="i") TYPE:e4a184d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||| N_EXP_INT_CONST (0,0)
||||| 0
||||| N_EXP_LSS (0,0)

```

```

||||||| N_EXP_IDENT (0,0)
||||||| (ID="i") TYPE:e4a184d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_INT_CONST (0,0)
||||||| 5
||||||| N_EXP_PRE_INC (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="i") TYPE:e4a184d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_STMT_COMPOUND (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_FUNCTION_CALL (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="printf") TYPE:e4a187a0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||||||| N_ARG_LIST (0,0)
||||||| N_EXP_STRING_LITERAL (0,0)
||||||| "%d\n"
||||||| N_ARG_LIST (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="i") TYPE:e4a184d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_ARG_LIST_NIL (0,0)
||||||| N_STMT_LIST_NIL (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_RETURN (0,0)
||||||| N_EXP_INT_CONST (0,0)
||||||| 0
||||||| N_STMT_LIST_NIL (0,0)
===== semantic tree =====
N_PROGRAM (0,20)
| (ID="main") TYPE:e4a1db60 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
| | TYPE
| | FUNCTION
| | | PARAMETER
| | | TYPE
| | | | (int)
| | | BODY
| | | | N_STMT_COMPOUND (0,4)
| | | | (ID="i") TYPE:e4a184d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
| | | | TYPE
| | | | | (int)
| | | | N_STMT_LIST (0,0)
| | | | N_STMT_FOR (0,0)
| | | | N_FOR_EXP (0,0)
| | | | | N_EXP_ASSIGN (e4a184d0,0)
| | | | | N_EXP_IDENT (e4a184d0,1)
| | | | | (ID="i") TYPE:e4a184d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
| | | | | N_EXP_INT_CONST (e4a184d0,0)
| | | | | INT=0
| | | | | N_EXP_LSS (e4a184d0,0)
| | | | | N_EXP_IDENT (e4a184d0,1)

```

```

||||||| (ID="i") TYPE:e4a184d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
||||||| N_EXP_INT_CONST (e4a184d0,0)
||||||| INT=5
||||||| N_EXP_PRE_INC (e4a184d0,0)
||||||| N_EXP_IDENT (e4a184d0,1)
||||||| (ID="i") TYPE:e4a184d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
||||||| N_STMT_COMPOUND (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_FUNCTION_CALL (e4a18680,0)
||||||| N_EXP_AMP (e4a1f360,0)
||||||| N_EXP_IDENT (e4a187a0,0)
||||||| (ID="printf") TYPE:e4a187a0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||||||| N_ARG_LIST (0,8)
||||||| N_EXP_STRING_LITERAL (e4a186c0,0)
||||||| LITERAL: "%d\n"
||||||| N_ARG_LIST (0,4)
||||||| N_EXP_IDENT (e4a184d0,1)
||||||| (ID="i") TYPE:e4a184d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
||||||| N_ARG_LIST_NIL (0,0)
||||||| N_STMT_LIST_NIL (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_RETURN (0,0)
||||||| N_EXP_INT_CONST (e4a184d0,0)
||||||| INT=0
||||||| N_STMT_LIST_NIL (0,0)

```

3) 덧셈 프로그램

```

===== syntax tree =====
N_PROGRAM (0,0)
| (ID="main") TYPE:f76dbb60 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
|| TYPE
|| FUNCTION
||| PARAMETER
||| TYPE
||| (int)
||| BODY
||| N_STMT_COMPOUND (0,0)
||| (ID="a") TYPE:f76d64d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||| TYPE
||| (int)
||| (ID="b") TYPE:f76d64d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||| TYPE
||| (int)
||| (ID="sum") TYPE:f76d64d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||| TYPE
||| (int)
||| N_STMT_LIST (0,0)

```

```

||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_ASSIGN (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="a") TYPE:f76d64d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_INT_CONST (0,0)
||||||| 100
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_ASSIGN (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="b") TYPE:f76d64d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_INT_CONST (0,0)
||||||| 200
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_ASSIGN (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="sum") TYPE:f76d64d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_ADD (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="a") TYPE:f76d64d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_IDENT (0,0)
||||||| (ID="b") TYPE:f76d64d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_FUNCTION_CALL (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="printf") TYPE:f76d67a0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||||||| N_ARG_LIST (0,0)
||||||| N_EXP_STRING_LITERAL (0,0)
||||||| "%d\n"
||||||| N_ARG_LIST (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="sum") TYPE:f76d64d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_ARG_LIST_NIL (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_RETURN (0,0)
||||||| N_EXP_INT_CONST (0,0)
||||||| 0
||||||| N_STMT_LIST_NIL (0,0)
===== semantic tree =====
N_PROGRAM (0,20)
| (ID="main") TYPE:f76dbb60 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
|| TYPE
|| FUNCTION
||| PARAMETER
||| TYPE
||| (int)
||| BODY

```

```

||||| N_STMT_COMPOUND (0,12)
|||||| (ID="a") TYPE:f76d64d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
|||||| TYPE
|||||| (int)
|||||| (ID="b") TYPE:f76d64d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=16
|||||| TYPE
|||||| (int)
|||||| (ID="sum") TYPE:f76d64d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=20
|||||| TYPE
|||||| (int)
||||| N_STMT_LIST (0,0)
||||| N_STMT_EXPRESSION (0,0)
||||| N_EXP_ASSIGN (f76d64d0,0)
||||| N_EXP_IDENT (f76d64d0,1)
|||||| (ID="a") TYPE:f76d64d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
|||||| N_EXP_INT_CONST (f76d64d0,0)
|||||| INT=100
||||| N_STMT_LIST (0,0)
||||| N_STMT_EXPRESSION (0,0)
||||| N_EXP_ASSIGN (f76d64d0,0)
||||| N_EXP_IDENT (f76d64d0,1)
|||||| (ID="b") TYPE:f76d64d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=16
|||||| N_EXP_INT_CONST (f76d64d0,0)
|||||| INT=200
||||| N_STMT_LIST (0,0)
||||| N_STMT_EXPRESSION (0,0)
||||| N_EXP_ASSIGN (f76d64d0,0)
||||| N_EXP_IDENT (f76d64d0,1)
|||||| (ID="sum") TYPE:f76d64d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=20
|||||| N_EXP_ADD (f76d64d0,0)
|||||| N_EXP_IDENT (f76d64d0,1)
|||||| (ID="a") TYPE:f76d64d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
|||||| N_EXP_IDENT (f76d64d0,1)
|||||| (ID="b") TYPE:f76d64d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=16
||||| N_STMT_LIST (0,0)
||||| N_STMT_EXPRESSION (0,0)
||||| N_EXP_FUNCTION_CALL (f76d6680,0)
||||| N_EXP_AMP (f76dd580,0)
||||| N_EXP_IDENT (f76d67a0,0)
|||||| (ID="printf") TYPE:f76d67a0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||||| N_ARG_LIST (0,8)
||||| N_EXP_STRING_LITERAL (f76d66c0,0)
|||||| LITERAL: "%d\n"
||||| N_ARG_LIST (0,4)
||||| N_EXP_IDENT (f76d64d0,1)
|||||| (ID="sum") TYPE:f76d64d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=20
|||||| N_ARG_LIST_NIL (0,0)
||||| N_STMT_LIST (0,0)
||||| N_STMT_RETURN (0,0)

```

```

||||||| N_EXP_INT_CONST (f76d64d0,0)
||||||| INT=0
||||||| N_STMT_LIST NIL (0,0)

```

4) scanf로 입력 받은 x,y값을 바꾸는 프로그램

```

===== syntax tree =====
N_PROGRAM (0,0)
| (ID="main") TYPE:3adf7b60 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
| | TYPE
| | | FUNCTION
| | | | PARAMETER
| | | | TYPE
| | | | | (int)
| | | | BODY
| | | | | N_STMT_COMPOUND (0,0)
| | | | | | (ID="x") TYPE:3adf24d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | TYPE
| | | | | | | | (int)
| | | | | | | | (ID="y") TYPE:3adf24d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | | TYPE
| | | | | | | | | (int)
| | | | | | | | | (ID="temp") TYPE:3adf24d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | | | TYPE
| | | | | | | | | | (int)
| | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | | | | N_EXP_FUNCTION_CALL (0,0)
| | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | (ID="scanf") TYPE:3adf28d0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
| | | | | | | | | | | N_ARG_LIST (0,0)
| | | | | | | | | | | N_EXP_STRING_LITERAL (0,0)
| | | | | | | | | | | "%d %d"
| | | | | | | | | | | N_ARG_LIST (0,0)
| | | | | | | | | | | N_EXP_AMP (0,0)
| | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | | (ID="x") TYPE:3adf24d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | | | | | | N_ARG_LIST (0,0)
| | | | | | | | | | | N_EXP_AMP (0,0)
| | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | | | (ID="y") TYPE:3adf24d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | | | | | | N_ARG_LIST NIL (0,0)
| | | | | | | | | | | N_STMT_LIST (0,0)
| | | | | | | | | | | N_STMT_EXPRESSION (0,0)
| | | | | | | | | | | N_EXP_ASSIGN (0,0)
| | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | | (ID="temp") TYPE:3adf24d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
| | | | | | | | | | | | N_EXP_IDENT (0,0)
| | | | | | | | | | | | | (ID="x") TYPE:3adf24d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0

```

```

||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_ASSIGN (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="x") TYPE:3adf24d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_IDENT (0,0)
||||||| (ID="y") TYPE:3adf24d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_ASSIGN (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="y") TYPE:3adf24d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_IDENT (0,0)
||||||| (ID="temp") TYPE:3adf24d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_FUNCTION_CALL (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="printf") TYPE:3adf27a0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||||||| N_ARG_LIST (0,0)
||||||| N_EXP_STRING_LITERAL (0,0)
||||||| "%d %d\n"
||||||| N_ARG_LIST (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="x") TYPE:3adf24d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_ARG_LIST (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="y") TYPE:3adf24d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_ARG_LIST_NIL (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_RETURN (0,0)
||||||| N_EXP_INT_CONST (0,0)
||||||| 0
||||||| N_STMT_LIST_NIL (0,0)
===== semantic tree =====
N_PROGRAM (0,32)
| (ID="main") TYPE:3adf7b60 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
|| TYPE
|| FUNCTION
||| PARAMETER
||| TYPE
||| (int)
||| BODY
||| N_STMT_COMPOUND (0,12)
||| (ID="x") TYPE:3adf24d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
||| TYPE
||| (int)
||| (ID="y") TYPE:3adf24d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=16
||| TYPE

```

```
||||||| (int)
||||||| (ID="temp") TYPE:3adf24d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=20
||||||| TYPE
||||||| (int)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_FUNCTION_CALL (3adf2680,0)
||||||| N_EXP_AMP (3adf98a0,0)
||||||| N_EXP_IDENT (3adf28d0,0)
||||||| (ID="scanf") TYPE:3adf28d0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||||||| N_ARG_LIST (0,12)
||||||| N_EXP_STRING_LITERAL (3adf26c0,0)
||||||| LITERAL: "%d %d"
||||||| N_ARG_LIST (0,8)
||||||| N_EXP_AMP (3adf9920,0)
||||||| N_EXP_IDENT (3adf24d0,1)
||||||| (ID="x") TYPE:3adf24d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
||||||| N_ARG_LIST (0,4)
||||||| N_EXP_AMP (3adf9960,0)
||||||| N_EXP_IDENT (3adf24d0,1)
||||||| (ID="y") TYPE:3adf24d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=16
||||||| N_ARG_LIST_NIL (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_ASSIGN (3adf24d0,0)
||||||| N_EXP_IDENT (3adf24d0,1)
||||||| (ID="temp") TYPE:3adf24d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=20
||||||| N_EXP_IDENT (3adf24d0,1)
||||||| (ID="x") TYPE:3adf24d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_ASSIGN (3adf24d0,0)
||||||| N_EXP_IDENT (3adf24d0,1)
||||||| (ID="x") TYPE:3adf24d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
||||||| N_EXP_IDENT (3adf24d0,1)
||||||| (ID="y") TYPE:3adf24d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=16
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_ASSIGN (3adf24d0,0)
||||||| N_EXP_IDENT (3adf24d0,1)
||||||| (ID="y") TYPE:3adf24d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=16
||||||| N_EXP_IDENT (3adf24d0,1)
||||||| (ID="temp") TYPE:3adf24d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=20
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_FUNCTION_CALL (3adf2680,0)
||||||| N_EXP_AMP (3adf99a0,0)
||||||| N_EXP_IDENT (3adf27a0,0)
||||||| (ID="printf") TYPE:3adf27a0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
```

```

||||||| N_ARG_LIST (0,12)
||||||| N_EXP_STRING_LITERAL (3adf26c0,0)
||||||| | LITERAL: "%d %d\n"
||||||| N_ARG_LIST (0,8)
||||||| | N_EXP_IDENT (3adf24d0,1)
||||||| | | (ID="x") TYPE:3adf24d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
||||||| N_ARG_LIST (0,4)
||||||| | N_EXP_IDENT (3adf24d0,1)
||||||| | | (ID="y") TYPE:3adf24d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=16
||||||| N_ARG_LIST_NIL (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_RETURN (0,0)
||||||| N_EXP_INT_CONST (3adf24d0,0)
||||||| | INT=0
||||||| N_STMT_LIST_NIL (0,0)

```

5) malloc한 뒤 배열의 값을 출력하는 프로그램

```

===== syntax tree =====
N_PROGRAM (0,0)
| (ID="main") TYPE:32829b60 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
|| TYPE
||| FUNCTION
||| PARAMETER
||| TYPE
|||| (void)||| BODY
|||| N_STMT_COMPOUND (0,0)
||||| (ID="a") TYPE:32829be0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||| TYPE
||||| POINTER
||||| ELEMENT_TYPE
||||| | (int)
||||| | (ID="count") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||| TYPE
||||| | (int)
||||| | (ID="i") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||| TYPE
||||| | (int)
||||| N_STMT_LIST (0,0)
||||| N_STMT_EXPRESSION (0,0)
||||| N_EXP_ASSIGN (0,0)
||||| N_EXP_IDENT (0,0)
||||| | (ID="count") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||| | N_EXP_INT_CONST (0,0)
||||| | | 10
||||| | N_STMT_LIST (0,0)
||||| | N_STMT_EXPRESSION (0,0)
||||| | N_EXP_ASSIGN (0,0)
||||| | N_EXP_IDENT (0,0)

```

```
||||||| (ID="a") TYPE:32829be0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_FUNCTION_CALL (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="malloc") TYPE:328249b0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||||||| N_ARG_LIST (0,0)
||||||| N_EXP_MUL (0,0)
||||||| N_EXP_SIZE_TYPE (0,0)
||||||| (int)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="count") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_ARG_LIST_NIL (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_FOR (0,0)
||||||| N_FOR_EXP (0,0)
||||||| N_EXP_ASSIGN (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="i") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_INT_CONST (0,0)
||||||| 0
||||||| N_EXP_LSS (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="i") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_IDENT (0,0)
||||||| (ID="count") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_PRE_INC (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="i") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_STMT_COMPOUND (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_ASSIGN (0,0)
||||||| N_EXP_ARRAY (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="a") TYPE:32829be0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_IDENT (0,0)
||||||| (ID="i") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_IDENT (0,0)
||||||| (ID="i") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_STMT_LIST_NIL (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_FOR (0,0)
||||||| N_FOR_EXP (0,0)
||||||| N_EXP_ASSIGN (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="i") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_INT_CONST (0,0)
||||||| 0
||||||| N_EXP_LSS (0,0)
||||||| N_EXP_IDENT (0,0)
```

```

||||||| (ID="i") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_IDENT (0,0)
||||||| (ID="count") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_PRE_INC (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="i") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_STMT_COMPOUND (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_FUNCTION_CALL (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="printf") TYPE:328247a0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||||||| N_ARG_LIST (0,0)
||||||| N_EXP_STRING_LITERAL (0,0)
||||||| "number[%d] : %d\n"
||||||| N_ARG_LIST (0,0)
||||||| N_EXP_ADD (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="i") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_INT_CONST (0,0)
||||||| 1
||||||| N_ARG_LIST (0,0)
||||||| N_EXP_STAR (0,0)
||||||| N_EXP_ADD (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="a") TYPE:32829be0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_IDENT (0,0)
||||||| (ID="i") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_ARG_LIST_NIL (0,0)
||||||| N_STMT_LIST_NIL (0,0)
||||||| N_STMT_LIST_NIL (0,0)
===== semantic tree =====
N_PROGRAM (0,32)
|(ID="main") TYPE:32829b60 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
|| TYPE
|| FUNCTION
||| PARAMETER
||| TYPE
||| (void)
||| BODY
|||| N_STMT_COMPOUND (0,12)
||||| (ID="a") TYPE:32829be0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
||||| TYPE
||||| POINTER
||||| ELEMENT_TYPE
||||| (int)
||||| (ID="count") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=16
||||| TYPE
||||| (int)

```

```
||||||| (ID="i") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=20
||||||| TYPE
||||||| (int)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_ASSIGN (328244d0,0)
||||||| N_EXP_IDENT (328244d0,1)
||||||| (ID="count") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=16
||||||| N_EXP_INT_CONST (328244d0,0)
||||||| INT=10
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_ASSIGN (32829be0,0)
||||||| N_EXP_IDENT (32829be0,1)
||||||| (ID="a") TYPE:32829be0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
||||||| N_EXP_FUNCTION_CALL (328246c0,0)
||||||| N_EXP_AMP (3282bee0,0)
||||||| N_EXP_IDENT (328249b0,0)
||||||| (ID="malloc") TYPE:328249b0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||||||| N_ARG_LIST (0,4)
||||||| N_EXP_MUL (328244d0,0)
||||||| N_EXP_SIZE_TYPE (328244d0,0)
||||||| INT=4
||||||| N_EXP_IDENT (328244d0,1)
||||||| (ID="count") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=16
||||||| N_ARG_LIST_NIL (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_FOR (0,0)
||||||| N_FOR_EXP (0,0)
||||||| N_EXP_ASSIGN (328244d0,0)
||||||| N_EXP_IDENT (328244d0,1)
||||||| (ID="i") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=20
||||||| N_EXP_INT_CONST (328244d0,0)
||||||| INT=0
||||||| N_EXP_LSS (328244d0,0)
||||||| N_EXP_IDENT (328244d0,1)
||||||| (ID="i") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=20
||||||| N_EXP_IDENT (328244d0,1)
||||||| (ID="count") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=16
||||||| N_EXP_PRE_INC (328244d0,0)
||||||| N_EXP_IDENT (328244d0,1)
||||||| (ID="i") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=20
||||||| N_STMT_COMPOUND (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_ASSIGN (328244d0,0)
||||||| N_EXP_ARRAY (328244d0,1)
||||||| N_EXP_IDENT (32829be0,1)
||||||| (ID="a") TYPE:32829be0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
```

```

||||||| N_EXP_IDENT (328244d0,1)
||||||| (ID="i") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=20
||||||| N_EXP_IDENT (328244d0,1)
||||||| (ID="i") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=20
||||||| N_STMT_LIST_NIL (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_FOR (0,0)
||||||| N_FOR_EXP (0,0)
||||||| N_EXP_ASSIGN (328244d0,0)
||||||| N_EXP_IDENT (328244d0,1)
||||||| (ID="i") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=20
||||||| N_EXP_INT_CONST (328244d0,0)
||||||| INT=0
||||||| N_EXP_LSS (328244d0,0)
||||||| N_EXP_IDENT (328244d0,1)
||||||| (ID="i") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=20
||||||| N_EXP_IDENT (328244d0,1)
||||||| (ID="count") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=16
||||||| N_EXP_PRE_INC (328244d0,0)
||||||| N_EXP_IDENT (328244d0,1)
||||||| (ID="i") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=20
||||||| N_STMT_COMPOUND (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_FUNCTION_CALL (32824680,0)
||||||| N_EXP_AMP (3282bf60,0)
||||||| N_EXP_IDENT (328247a0,0)
||||||| (ID="printf") TYPE:328247a0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||||||| N_ARG_LIST (0,12)
||||||| N_EXP_STRING_LITERAL (328246c0,0)
||||||| LITERAL: "number[%d] : %d\n"
||||||| N_ARG_LIST (0,8)
||||||| N_EXP_ADD (328244d0,0)
||||||| N_EXP_IDENT (328244d0,1)
||||||| (ID="i") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=20
||||||| N_EXP_INT_CONST (328244d0,0)
||||||| INT=1
||||||| N_ARG_LIST (0,4)
||||||| N_EXP_STAR (328244d0,1)
||||||| N_EXP_ADD (32829be0,0)
||||||| N_EXP_IDENT (32829be0,1)
||||||| (ID="a") TYPE:32829be0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
||||||| N_EXP_IDENT (328244d0,1)
||||||| (ID="i") TYPE:328244d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=20
||||||| N_ARG_LIST_NIL (0,0)
||||||| N_STMT_LIST_NIL (0,0)
||||||| N_STMT_LIST_NIL (0,0)

```

6) scanf로 입력 받은 두 값을 사칙연산 등 여러가지 결과를 출력하는 프로그램

```

===== syntax tree =====
N_PROGRAM (0,0)
| (ID="main") TYPE:e9c7db60 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
|| TYPE
||| FUNCTION
||| | PARAMETER
||| | TYPE
||| | | (int)
||| | BODY
||| | | N_STMT_COMPOUND (0,0)
||| | | | (ID="a") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||| | | | TYPE
||| | | | | (int)
||| | | | | (ID="b") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||| | | | TYPE
||| | | | | (int)
||| | | | N_STMT_LIST (0,0)
||| | | | | N_STMT_EXPRESSION (0,0)
||| | | | | N_EXP_FUNCTION_CALL (0,0)
||| | | | | N_EXP_IDENT (0,0)
||| | | | | (ID="scanf") TYPE:e9c788d0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||| | | | | N_ARG_LIST (0,0)
||| | | | | N_EXP_STRING_LITERAL (0,0)
||| | | | | "%d %d"
||| | | | | N_ARG_LIST (0,0)
||| | | | | N_EXP_AMP (0,0)
||| | | | | N_EXP_IDENT (0,0)
||| | | | | | (ID="a") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||| | | | | | N_ARG_LIST (0,0)
||| | | | | | N_EXP_AMP (0,0)
||| | | | | | N_EXP_IDENT (0,0)
||| | | | | | | (ID="b") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||| | | | | | | N_ARG_LIST_NIL (0,0)
||| | | | | N_STMT_LIST (0,0)
||| | | | | N_STMT_EXPRESSION (0,0)
||| | | | | N_EXP_FUNCTION_CALL (0,0)
||| | | | | N_EXP_IDENT (0,0)
||| | | | | (ID="printf") TYPE:e9c787a0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||| | | | | N_ARG_LIST (0,0)
||| | | | | N_EXP_STRING_LITERAL (0,0)
||| | | | | "result : %d\n"
||| | | | | N_ARG_LIST (0,0)
||| | | | | N_EXP_ADD (0,0)
||| | | | | N_EXP_IDENT (0,0)
||| | | | | | (ID="a") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||| | | | | | N_EXP_IDENT (0,0)
||| | | | | | | (ID="b") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||| | | | | | | N_ARG_LIST_NIL (0,0)

```

```
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_FUNCTION_CALL (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="printf") TYPE:e9c787a0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||||||| N_ARG_LIST (0,0)
||||||| N_EXP_STRING_LITERAL (0,0)
||||||| "result : %d\n"
||||||| N_ARG_LIST (0,0)
||||||| N_EXP_SUB (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="a") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_IDENT (0,0)
||||||| (ID="b") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_ARG_LIST_NIL (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_FUNCTION_CALL (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="printf") TYPE:e9c787a0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||||||| N_ARG_LIST (0,0)
||||||| N_EXP_STRING_LITERAL (0,0)
||||||| "result : %d\n"
||||||| N_ARG_LIST (0,0)
||||||| N_EXP_MUL (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="a") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_IDENT (0,0)
||||||| (ID="b") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_ARG_LIST_NIL (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_FUNCTION_CALL (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="printf") TYPE:e9c787a0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||||||| N_ARG_LIST (0,0)
||||||| N_EXP_STRING_LITERAL (0,0)
||||||| "result : %d\n"
||||||| N_ARG_LIST (0,0)
||||||| N_EXP_DIV (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="a") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_IDENT (0,0)
||||||| (ID="b") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_ARG_LIST_NIL (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_FUNCTION_CALL (0,0)
||||||| N_EXP_IDENT (0,0)
```

```

||||||| (ID="printf") TYPE:e9c787a0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||||||| N_ARG_LIST (0,0)
||||||| N_EXP_STRING_LITERAL (0,0)
||||||| "result : %d\n"
||||||| N_ARG_LIST (0,0)
||||||| N_EXP_MOD (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="a") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_IDENT (0,0)
||||||| (ID="b") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_ARG_LIST_NIL (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_RETURN (0,0)
||||||| N_EXP_INT_CONST (0,0)
||||||| 0
||||||| N_STMT_LIST_NIL (0,0)
===== semantic tree =====
N_PROGRAM (0,100)
| (ID="main") TYPE:e9c7db60 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
| TYPE
| FUNCTION
| PARAMETER
| TYPE
| | (int)
| BODY
| N_STMT_COMPOUND (0,8)
| | (ID="a") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
| | TYPE
| | | (int)
| | | (ID="b") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=16
| | TYPE
| | | (int)
| | N_STMT_LIST (0,0)
| | N_STMT_EXPRESSION (0,0)
| | N_EXP_FUNCTION_CALL (e9c78680,0)
| | N_EXP_AMP (e9c80010,0)
| | N_EXP_IDENT (e9c788d0,0)
| | (ID="scanf") TYPE:e9c788d0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
| | N_ARG_LIST (0,12)
| | | N_EXP_STRING_LITERAL (e9c786c0,0)
| | | LITERAL: "%d %d"
| | | N_ARG_LIST (0,8)
| | | N_EXP_AMP (e9c80090,0)
| | | N_EXP_IDENT (e9c784d0,1)
| | | (ID="a") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
| | | N_ARG_LIST (0,4)
| | | N_EXP_AMP (e9c800d0,0)
| | | N_EXP_IDENT (e9c784d0,1)
| | | (ID="b") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=16

```

```
||||||| N_ARG_LIST_NIL (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_FUNCTION_CALL (e9c78680,0)
||||||| N_EXP_AMP (e9c80110,0)
||||||| N_EXP_IDENT (e9c787a0,0)
||||||| (ID="printf") TYPE:e9c787a0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||||||| N_ARG_LIST (0,8)
||||||| N_EXP_STRING_LITERAL (e9c786c0,0)
||||||| LITERAL: "result : %d\n"
||||||| N_ARG_LIST (0,4)
||||||| N_EXP_ADD (e9c784d0,0)
||||||| N_EXP_IDENT (e9c784d0,1)
||||||| (ID="a") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
||||||| N_EXP_IDENT (e9c784d0,1)
||||||| (ID="b") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=16
||||||| N_ARG_LIST_NIL (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_FUNCTION_CALL (e9c78680,0)
||||||| N_EXP_AMP (e9c80190,0)
||||||| N_EXP_IDENT (e9c787a0,0)
||||||| (ID="printf") TYPE:e9c787a0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||||||| N_ARG_LIST (0,8)
||||||| N_EXP_STRING_LITERAL (e9c786c0,0)
||||||| LITERAL: "result : %d\n"
||||||| N_ARG_LIST (0,4)
||||||| N_EXP_SUB (e9c784d0,0)
||||||| N_EXP_IDENT (e9c784d0,1)
||||||| (ID="a") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
||||||| N_EXP_IDENT (e9c784d0,1)
||||||| (ID="b") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=16
||||||| N_ARG_LIST_NIL (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_FUNCTION_CALL (e9c78680,0)
||||||| N_EXP_AMP (e9c80210,0)
||||||| N_EXP_IDENT (e9c787a0,0)
||||||| (ID="printf") TYPE:e9c787a0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||||||| N_ARG_LIST (0,8)
||||||| N_EXP_STRING_LITERAL (e9c786c0,0)
||||||| LITERAL: "result : %d\n"
||||||| N_ARG_LIST (0,4)
||||||| N_EXP_MUL (e9c784d0,0)
||||||| N_EXP_IDENT (e9c784d0,1)
||||||| (ID="a") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
||||||| N_EXP_IDENT (e9c784d0,1)
||||||| (ID="b") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=16
||||||| N_ARG_LIST_NIL (0,0)
```

```

||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_FUNCTION_CALL (e9c78680,0)
||||||| N_EXP_AMP (e9c80290,0)
||||||| N_EXP_IDENT (e9c787a0,0)
||||||| | (ID="printf") TYPE:e9c787a0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||||||| N_ARG_LIST (0,8)
||||||| N_EXP_STRING_LITERAL (e9c786c0,0)
||||||| | LITERAL: "result : %d\n"
||||||| N_ARG_LIST (0,4)
||||||| N_EXP_DIV (e9c784d0,0)
||||||| N_EXP_IDENT (e9c784d0,1)
||||||| | (ID="a") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
||||||| | N_EXP_IDENT (e9c784d0,1)
||||||| | | (ID="b") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=16
||||||| N_ARG_LIST NIL (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_FUNCTION_CALL (e9c78680,0)
||||||| N_EXP_AMP (e9c80310,0)
||||||| N_EXP_IDENT (e9c787a0,0)
||||||| | (ID="printf") TYPE:e9c787a0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||||||| N_ARG_LIST (0,8)
||||||| N_EXP_STRING_LITERAL (e9c786c0,0)
||||||| | LITERAL: "result : %d\n"
||||||| N_ARG_LIST (0,4)
||||||| N_EXP_MOD (e9c784d0,0)
||||||| N_EXP_IDENT (e9c784d0,1)
||||||| | (ID="a") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
||||||| | N_EXP_IDENT (e9c784d0,1)
||||||| | | (ID="b") TYPE:e9c784d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=16
||||||| N_ARG_LIST NIL (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_RETURN (0,0)
||||||| N_EXP_INT_CONST (e9c784d0,0)
||||||| | INT=0
||||||| N_STMT_LIST NIL (0,0)

```

7) sort 프로그램

```

===== syntax tree =====
N_PROGRAM (0,0)
| (ID="a") TYPE:eaee2ba0 KIND:VAR SPEC=AUTO LEV=0 VAL=0 ADDR=0
|| TYPE
|| | ARRAY
|| | | INDEX
|| | | N_EXP_INT_CONST (0,0)
|| | | | 5
(none)

```

```
|||||ELEMENT_TYPE
|||||(int)
| (ID="sort") TYPE:eaee2db0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
|| TYPE
|| FUNCTION
||| PARAMETER
|||||(ID="a") TYPE:eaee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=0
||||| TYPE
||||| ARRAY
||||| INDEX
||||||(none)
||||| ELEMENT_TYPE
|||||(int)
|||||(ID="n") TYPE:eaedd4d0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=0
||||| TYPE
|||||(int)
||||| TYPE
|||||(void)||| BODY
||||| N_STMT_COMPOUND (0,0)
|||||(ID="x") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||| TYPE
|||||(int)
|||||(ID="y") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||| TYPE
|||||(int)
|||||(ID="z") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||| TYPE
|||||(int)
||||| N_STMT_LIST (0,0)
||||| N_STMT_FOR (0,0)
||||| N_FOR_EXP (0,0)
||||||| N_EXP_ASSIGN (0,0)
||||||| N_EXP_IDENT (0,0)
||||||(ID="x") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_INT_CONST (0,0)
||||||| 0
||||||| N_EXP_LSS (0,0)
||||||| N_EXP_IDENT (0,0)
||||||(ID="x") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_IDENT (0,0)
||||||(ID="n") TYPE:eaedd4d0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=0
||||||| N_EXP_POST_INC (0,0)
||||||| N_EXP_IDENT (0,0)
||||||(ID="x") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_STMT_FOR (0,0)
||||||| N_FOR_EXP (0,0)
||||||| N_EXP_ASSIGN (0,0)
||||||| N_EXP_IDENT (0,0)
||||||(ID="y") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
```

```
||||||| N_EXP_ADD (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="x") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_INT_CONST (0,0)
||||||| 1
||||||| N_EXP_LSS (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="y") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_IDENT (0,0)
||||||| (ID="n") TYPE:eaedd4d0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=0
||||||| N_EXP_ASSIGN (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="y") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_ADD (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="y") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_INT_CONST (0,0)
||||||| 1
||||||| N_STMT_IF (0,0)
||||||| N_EXP_LSS (0,0)
||||||| N_EXP_ARRAY (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="a") TYPE:eaeee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=0
||||||| N_EXP_IDENT (0,0)
||||||| (ID="x") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_ARRAY (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="a") TYPE:eaeee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=0
||||||| N_EXP_IDENT (0,0)
||||||| (ID="y") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_STMT_COMPOUND (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_ASSIGN (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="z") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_ARRAY (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="a") TYPE:eaeee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=0
||||||| N_EXP_IDENT (0,0)
||||||| (ID="x") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_ASSIGN (0,0)
||||||| N_EXP_ARRAY (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="a") TYPE:eaeee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=0
||||||| N_EXP_IDENT (0,0)
||||||| (ID="x") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
```

```
||||||| N_EXP_ARRAY (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="a") TYPE:eaeee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=0
||||||| N_EXP_IDENT (0,0)
||||||| (ID="y") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_ASSIGN (0,0)
||||||| N_EXP_ARRAY (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="a") TYPE:eaeee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=0
||||||| N_EXP_IDENT (0,0)
||||||| (ID="y") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_IDENT (0,0)
||||||| (ID="z") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_STMT_LIST_NIL (0,0)
||||||| N_STMT_LIST_NIL (0,0)
|(ID="main") TYPE:eaeee3f30 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
|| TYPE
|| FUNCTION
||| PARAMETER
||| TYPE
||| (int)
||| BODY
|||| N_STMT_COMPOUND (0,0)
|||| (ID="i") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||| TYPE
||||| (int)
||||| N_STMT_LIST (0,0)
||||| N_STMT_EXPRESSION (0,0)
||||| N_EXP_ASSIGN (0,0)
||||| N_EXP_ARRAY (0,0)
||||| N_EXP_IDENT (0,0)
||||| (ID="a") TYPE:eaeee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=0
||||| N_EXP_INT_CONST (0,0)
||||| 0
||||| N_EXP_INT_CONST (0,0)
||||| 3
||||| N_STMT_LIST (0,0)
||||| N_STMT_EXPRESSION (0,0)
||||| N_EXP_ASSIGN (0,0)
||||| N_EXP_ARRAY (0,0)
||||| N_EXP_IDENT (0,0)
||||| (ID="a") TYPE:eaeee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=0
||||| N_EXP_INT_CONST (0,0)
||||| 1
||||| N_EXP_INT_CONST (0,0)
||||| 1
||||| N_STMT_LIST (0,0)
```

```
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_ASSIGN (0,0)
||||||| N_EXP_ARRAY (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="a") TYPE:eaeee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=0
||||||| N_EXP_INT_CONST (0,0)
||||||| 2
||||||| N_EXP_INT_CONST (0,0)
||||||| 4
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_ASSIGN (0,0)
||||||| N_EXP_ARRAY (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="a") TYPE:eaeee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=0
||||||| N_EXP_INT_CONST (0,0)
||||||| 3
||||||| N_EXP_INT_CONST (0,0)
||||||| 2
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_ASSIGN (0,0)
||||||| N_EXP_ARRAY (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="a") TYPE:eaeee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=0
||||||| N_EXP_INT_CONST (0,0)
||||||| 4
||||||| N_EXP_INT_CONST (0,0)
||||||| 5
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_FUNCTION_CALL (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="sort") TYPE:eaeee2db0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||||||| N_ARG_LIST (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="a") TYPE:eaeee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=0
||||||| N_ARG_LIST (0,0)
||||||| N_EXP_INT_CONST (0,0)
||||||| 5
||||||| N_ARG_LIST_NIL (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_FOR (0,0)
||||||| N_FOR_EXP (0,0)
||||||| N_EXP_ASSIGN (0,0)
||||||| N_EXP_IDENT (0,0)
||||||| (ID="i") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||||| N_EXP_INT_CONST (0,0)
||||||| 0
```

```

||||| N_EXP_LSS (0,0)
||||| N_EXP_IDENT (0,0)
||| (ID="i") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||| N_EXP_INT_CONST (0,0)
||| 5
||||| N_EXP_POST_INC (0,0)
||||| N_EXP_IDENT (0,0)
||| (ID="i") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||| N_STMT_COMPOUND (0,0)
||||| N_STMT_LIST (0,0)
||||| N_STMT_EXPRESSION (0,0)
||||| N_EXP_FUNCTION_CALL (0,0)
||||| N_EXP_IDENT (0,0)
||| (ID="printf") TYPE:eaedd7a0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||||| N_ARG_LIST (0,0)
||||| N_EXP_STRING_LITERAL (0,0)
||||| "%d "
||||| N_ARG_LIST (0,0)
||||| N_EXP_ARRAY (0,0)
||||| N_EXP_IDENT (0,0)
||| (ID="a") TYPE:eaee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=0
||||| N_EXP_IDENT (0,0)
||| (ID="i") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=0
||||| N_ARG_LIST_NIL (0,0)
||||| N_STMT_LIST_NIL (0,0)
||||| N_STMT_LIST (0,0)
||||| N_STMT_EXPRESSION (0,0)
||||| N_EXP_FUNCTION_CALL (0,0)
||||| N_EXP_IDENT (0,0)
||| (ID="printf") TYPE:eaedd7a0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||||| N_ARG_LIST (0,0)
||||| N_EXP_STRING_LITERAL (0,0)
||||| "\n"
||||| N_ARG_LIST_NIL (0,0)
||||| N_STMT_LIST_NIL (0,0)
===== semantic tree =====
N_PROGRAM (0,48)
| (ID="a") TYPE:eaee2ba0 KIND:VAR SPEC=AUTO LEV=0 VAL=0 ADDR=12
|| TYPE
|| ARRAY
||| INDEX
||| INT=5
||| ELEMENT_TYPE
||| (int)
| (ID="sort") TYPE:eaee2db0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
|| TYPE
|| FUNCTION
||| PARAMETER
||| (ID="a") TYPE:eaee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=12

```

```

||||||| TYPE
||||||| POINTER
||||||| ELEMENT_TYPE
||||||| (int)
||||| (ID="n") TYPE:eaedd4d0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=16
||||||| TYPE
||||||| (int)
||||| TYPE
||||| (void)
|||| BODY
||||| N_STMT_COMPOUND (0,12)
||||| (ID="x") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=20
||||||| TYPE
||||||| (int)
||||| (ID="y") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=24
||||||| TYPE
||||||| (int)
||||| (ID="z") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=28
||||||| TYPE
||||||| (int)
||||| N_STMT_LIST (0,0)
||||| N_STMT_FOR (0,0)
||||||| N_FOR_EXP (0,0)
||||||| N_EXP_ASSIGN (eaedd4d0,0)
||||||| N_EXP_IDENT (eaedd4d0,1)
||||||| (ID="x") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=20
||||||| N_EXP_INT_CONST (eaedd4d0,0)
||||||| INT=0
||||||| N_EXP_LSS (eaedd4d0,0)
||||||| N_EXP_IDENT (eaedd4d0,1)
||||||| (ID="x") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=20
||||||| N_EXP_IDENT (eaedd4d0,1)
||||||| (ID="n") TYPE:eaedd4d0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=16
||||||| N_EXP_POST_INC (eaedd4d0,0)
||||||| N_EXP_IDENT (eaedd4d0,1)
||||||| (ID="x") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=20
||||||| N_STMT_FOR (0,0)
||||||| N_FOR_EXP (0,0)
||||||| N_EXP_ASSIGN (eaedd4d0,0)
||||||| N_EXP_IDENT (eaedd4d0,1)
||||||| (ID="y") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=24
||||||| N_EXP_ADD (eaedd4d0,0)
||||||| N_EXP_IDENT (eaedd4d0,1)
||||||| (ID="x") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=20
||||||| N_EXP_INT_CONST (eaedd4d0,0)
||||||| INT=1
||||||| N_EXP_LSS (eaedd4d0,0)
||||||| N_EXP_IDENT (eaedd4d0,1)
||||||| (ID="y") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=24

```

```
||||||| N_EXP_IDENT (eaedd4d0,1)
||||||| (ID="n") TYPE:eaedd4d0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=16
||||||| N_EXP_ASSIGN (eaedd4d0,0)
||||||| N_EXP_IDENT (eaedd4d0,1)
||||||| (ID="y") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=24
||||||| N_EXP_ADD (eaedd4d0,0)
||||||| N_EXP_IDENT (eaedd4d0,1)
||||||| (ID="y") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=24
||||||| N_EXP_INT_CONST (eaedd4d0,0)
||||||| INT=1
||||||| N_STMT_IF (0,0)
||||||| N_EXP_LSS (eaedd4d0,0)
||||||| N_EXP_ARRAY (eaedd4d0,1)
||||||| N_EXP_IDENT (eaee2ce0,1)
||||||| (ID="a") TYPE:eaee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=12
||||||| N_EXP_IDENT (eaedd4d0,1)
||||||| (ID="x") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=20
||||||| N_EXP_ARRAY (eaedd4d0,1)
||||||| N_EXP_IDENT (eaee2ce0,1)
||||||| (ID="a") TYPE:eaee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=12
||||||| N_EXP_IDENT (eaedd4d0,1)
||||||| (ID="y") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=24
||||||| N_STMT_COMPOUND (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_ASSIGN (eaedd4d0,0)
||||||| N_EXP_IDENT (eaedd4d0,1)
||||||| (ID="z") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=28
||||||| N_EXP_ARRAY (eaedd4d0,1)
||||||| N_EXP_IDENT (eaee2ce0,1)
||||||| (ID="a") TYPE:eaee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=12
||||||| N_EXP_IDENT (eaedd4d0,1)
||||||| (ID="x") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=20
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_ASSIGN (eaedd4d0,0)
||||||| N_EXP_ARRAY (eaedd4d0,1)
||||||| N_EXP_IDENT (eaee2ce0,1)
||||||| (ID="a") TYPE:eaee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=12
||||||| N_EXP_IDENT (eaedd4d0,1)
||||||| (ID="x") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=20
||||||| N_EXP_ARRAY (eaedd4d0,1)
||||||| N_EXP_IDENT (eaee2ce0,1)
||||||| (ID="a") TYPE:eaee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=12
||||||| N_EXP_IDENT (eaedd4d0,1)
||||||| (ID="y") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=24
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_ASSIGN (eaedd4d0,0)
```

```
||||||| N_EXP_ARRAY (eaedd4d0,1)
||||||| N_EXP_IDENT (eaee2ce0,1)
||||||| (ID="a") TYPE:eaee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=12
||||||| N_EXP_IDENT (eaedd4d0,1)
||||||| (ID="y") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=24
||||||| N_EXP_IDENT (eaedd4d0,1)
||||||| (ID="z") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=28
||||||| N_STMT_LIST_NIL (0,0)
||||||| N_STMT_LIST_NIL (0,0)
| (ID="main") TYPE:eaee3f30 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
|| TYPE
|| FUNCTION
||| PARAMETER
||| TYPE
||| (int)
||| BODY
||| N_STMT_COMPOUND (0,4)
||| (ID="i") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
||| TYPE
||| (int)
||| N_STMT_LIST (0,0)
||| N_STMT_EXPRESSION (0,0)
||| N_EXP_ASSIGN (eaedd4d0,0)
||| N_EXP_ARRAY (eaedd4d0,1)
||| N_EXP_IDENT (eaee2ce0,1)
||| (ID="a") TYPE:eaee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=12
||| N_EXP_INT_CONST (eaedd4d0,0)
||| INT=0
||| N_EXP_INT_CONST (eaedd4d0,0)
||| INT=3
||| N_STMT_LIST (0,0)
||| N_STMT_EXPRESSION (0,0)
||| N_EXP_ASSIGN (eaedd4d0,0)
||| N_EXP_ARRAY (eaedd4d0,1)
||| N_EXP_IDENT (eaee2ce0,1)
||| (ID="a") TYPE:eaee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=12
||| N_EXP_INT_CONST (eaedd4d0,0)
||| INT=1
||| N_EXP_INT_CONST (eaedd4d0,0)
||| INT=1
||| N_STMT_LIST (0,0)
||| N_STMT_EXPRESSION (0,0)
||| N_EXP_ASSIGN (eaedd4d0,0)
||| N_EXP_ARRAY (eaedd4d0,1)
||| N_EXP_IDENT (eaee2ce0,1)
||| (ID="a") TYPE:eaee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=12
||| N_EXP_INT_CONST (eaedd4d0,0)
||| INT=2
||| N_EXP_INT_CONST (eaedd4d0,0)
```

```
||||||| INT=4
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_ASSIGN (eaedd4d0,0)
||||||| N_EXP_ARRAY (eaedd4d0,1)
||||||| N_EXP_IDENT (eaee2ce0,1)
||||||| (ID="a") TYPE:eaeee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=12
||||||| N_EXP_INT_CONST (eaedd4d0,0)
||||||| INT=3
||||||| N_EXP_INT_CONST (eaedd4d0,0)
||||||| INT=2
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_ASSIGN (eaedd4d0,0)
||||||| N_EXP_ARRAY (eaedd4d0,1)
||||||| N_EXP_IDENT (eaee2ce0,1)
||||||| (ID="a") TYPE:eaeee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=12
||||||| N_EXP_INT_CONST (eaedd4d0,0)
||||||| INT=4
||||||| N_EXP_INT_CONST (eaedd4d0,0)
||||||| INT=5
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_FUNCTION_CALL (eaedd680,0)
||||||| N_EXP_AMP (eaee63d0,0)
||||||| N_EXP_IDENT (eaee2db0,0)
||||||| (ID="sort") TYPE:eaeee2db0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||||||| N_ARG_LIST (0,8)
||||||| N_EXP_IDENT (eaee2ce0,1)
||||||| (ID="a") TYPE:eaeee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=12
||||||| N_ARG_LIST (0,4)
||||||| N_EXP_INT_CONST (eaedd4d0,0)
||||||| INT=5
||||||| N_ARG_LIST_NIL (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_FOR (0,0)
||||||| N_FOR_EXP (0,0)
||||||| N_EXP_ASSIGN (eaedd4d0,0)
||||||| N_EXP_IDENT (eaedd4d0,1)
||||||| (ID="i") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
||||||| N_EXP_INT_CONST (eaedd4d0,0)
||||||| INT=0
||||||| N_EXP_LSS (eaedd4d0,0)
||||||| N_EXP_IDENT (eaedd4d0,1)
||||||| (ID="i") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
||||||| N_EXP_INT_CONST (eaedd4d0,0)
||||||| INT=5
||||||| N_EXP_POST_INC (eaedd4d0,0)
||||||| N_EXP_IDENT (eaedd4d0,1)
```

```

||||||| (ID="i") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
||||||| N_STMT_COMPOUND (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_FUNCTION_CALL (eaedd680,0)
||||||| N_EXP_AMP (eaee6450,0)
||||||| N_EXP_IDENT (eaedd7a0,0)
||||||| (ID="printf") TYPE:eaedd7a0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||||||| N_ARG_LIST (0,8)
||||||| N_EXP_STRING_LITERAL (eaedd6c0,0)
||||||| LITERAL: "%d "
||||||| N_ARG_LIST (0,4)
||||||| N_EXP_ARRAY (eaedd4d0,1)
||||||| N_EXP_IDENT (eaee2ce0,1)
||||||| (ID="a") TYPE:eaee2ce0 KIND:PARM SPEC=NULL LEV=1 VAL=0 ADDR=12
||||||| N_EXP_IDENT (eaedd4d0,1)
||||||| (ID="i") TYPE:eaedd4d0 KIND:VAR SPEC=AUTO LEV=1 VAL=0 ADDR=12
||||||| N_ARG_LIST_NIL (0,0)
||||||| N_STMT_LIST_NIL (0,0)
||||||| N_STMT_LIST (0,0)
||||||| N_STMT_EXPRESSION (0,0)
||||||| N_EXP_FUNCTION_CALL (eaedd680,0)
||||||| N_EXP_AMP (eaee64d0,0)
||||||| N_EXP_IDENT (eaedd7a0,0)
||||||| (ID="printf") TYPE:eaedd7a0 KIND:FUNC SPEC=NULL LEV=0 VAL=0 ADDR=0
||||||| N_ARG_LIST (0,4)
||||||| N_EXP_STRING_LITERAL (eaedd6c0,0)
||||||| LITERAL: "\n"
||||||| N_ARG_LIST_NIL (0,0)
||||||| N_STMT_LIST_NIL (0,0)

```

4. 소스 파일 및 디렉토리 구조 참고 사항

1) 소스파일 : 별도 첨부

2) 디렉토리 구조

interp : 교수님께서 주신 첨부 파일 및 절차대로 설치

update: generation.c 추가작성

기존 코드에서 Makefile, kim.y, semantic_analysis.c 일부 코드 수정

test: interp폴더의 interp.out(i.out)과 update폴더의 a.out을 통한 테스트 작업 수행

끝.