15.1 4 1 -1

15.2 24 288 27648

15.3

x = 1, y = 2

x = 2, y = 1

x = 1, y = 2

15.9

```
1     #include <stdio.h>
2
3  +  int Func1(int, int);
4  +  int Func2(int);
5
6     int main() {
7         int x = 1;
8         int y = 2;
9
10        x = Func1(x, y);
11        y = Func2(y);
12
13        printf("x = %d y = %d\n", x, y);
14    }
15
16 -  int Func1(int x) {
17 +  int Func1(int x, int y) {
18        return x + y;
19    }
20
21    int Func2(int x) {
22        int y;
23 ?      //此处的 y 没有初始化，但不知 Func2 的具体功能，因而无法判断.
24        return x - y;
25    }
```

15.7

```
ToUpper:  SUBI  R29, R29, #4
          SW    0(R29), R30
          ADDI  R30, R29, #4
          SUBI  R29, R29, #4
          SW    0(R29), R16
          ;
          SLTI  R8, R4, x61
          BNEZ  R8, J1
          SLEI  R8, R4, x7A
          BEQZ  R8, J1
          SUBI  R4, R4, x20
J1:       ADDI  R16, R4, #0
          ADDI  R2, R16, #0
          ;
          LW    R16, 0(R29)
          ADDI  R29, R29, #4
          LW    R30, 0(R29)
          ADDI  R29, R29, #4
          RET
```

16.1

1). 5

2). 运行时栈中有三个变量 x, ptr1, ptr2，其中，ptr2 的值为 ptr1 的地址，ptr1 的值为 x 的地址，x 的值为 5.

16.2

1). 9

2). HELLO

16.9

```
StringLength:   SUBI  R29, R29, #4
                SW    0(R29), R16
                ;
                ADDI  R16, R0, #0
J1:             ADDI  R8, R4, R16
                LB    R9, 0(R8)
                BEQZ  R9, J2
                ADDI  R16, R16, #1
                J     J1
J2:             ADDI  R2, R16, #0
                ;
                LW    R16, 0(R29)
                ADDI  R29, R29, #4
                RET
```

16.10

```
1      #include <stdio.h>
2
3      char* ToUpper(char* inchar);
4
5      int main() {
6          char str[10];
7
8          printf("Enter a string: ");
9          scanf("%s", str);
10
11         printf("%s\n", ToUpper(str));
12     }
13
14     char* ToUpper(char* inchar) {
15 -       char str[10];
16 +       char* str = inchar;
17         int i = 0;
18         while (*(inchar + i) != '\0') {
19             if ('a' <= *(inchar + i) && *(inchar + i) <= 'z')
20                 *(str + i) = *(inchar + i) - ('a' - 'A');
21             else
22                 *(str + i) = *(inchar + i);
23             i++;
24         }
25         return str;
26     }
```

16.14 abc123