SORU9 :

Precedence	Highest	*, /, not
		+, −, &, mod
		− (unary)
		=, /=, <, <=, >, >=
		and
	Lowest	or, xor
Associativity	left to right	

a) $a * b - 1 + c$

$\Rightarrow (a * b)^1$

$((a * b)^1 - 1)^2$

$(((a * b)^1 - 1)^2 + c)^3$

b) $a * (b - 1) / c \bmod d$

$\Rightarrow (b - 1)^1$

$(a * (b - 1)^1)^2$

$((a * (b - 1)^1)^2 / c)^3$

$(((a * (b - 1)^1)^2 / c)^3 \bmod d)^4$

c) $(a - b) / c \ \& \ (d * e / a - 3)$

$\Rightarrow (a - b)^1 \quad (d * e)^2$

$((d * e)^2 / a)^3$

$(((d * e)^2 / a)^3 - 3)^4$

$((a - b)^1 / c)^5$

$(((a - b)^1 / c)^5 \ \& \ (((d * e)^2 / a)^3 - 3)^4)^6$

d) $-a \text{ or } c = d \text{ and } e$

$\Rightarrow (-a)^1$

$(c = d)^2$

$((c = d)^2 \text{ and } e)^3$

$((-a)^1 \text{ or } ((c = d)^2 \text{ and } e)^3)^4$

e) a > b xor c or d <= 17

⇒ (a > b)[1]

(d <= 17)[2]

((a > b)[1] xor c)[3]

(((a > b)[1] xor c)[3] or (d <= 17)[2])[4]

f) -a + b

⇒ (-a)[1]

((-a)[1] + b)[2]

SORU 13 :

```
int fun(int *k){

    *k += 4;
    return 3*(*k) - 1;
}
void main(){
    int i = 10, j = 10, sum1, sum2;
    sum1 = (i/2) + fun(&i);
    sum2 = fun(&j) + (j/2);
}
```

a) left to right

sum1 = (i/2) + fun(&i);

→ L2

* önce (i/2) yapılır.
  10/2 = 5
* sonra i=10 fun'a gönderilir.
  i += 4 i = 14
  14*3 - 1 = 41
* (i/2) + fun(&i)
  5 + 41 = 46

sum 2 = fun(&j) + (j/2);

→ L2

* önce j=10 fun'a gönderilir.
  j += 4 j = 14 olur.
  (14*3) - 1 = 41
* sonra (j/2) yapılır. j 14 olmuştu.
  14/2 = 7
* fun(&j) + (j/2)
  41 + 7 = 48

b) Right to left

sum1 = (i/2) + fun(8i);
RL ←

- Önce fun(8i) yapılır.
  - i = 10
  - 10+4 = 14 = i
  - (14×3) - 1 = 41
- sonra i/2 yapılır.
  - i = 14
  - 14/2 = 7
- toplama işlemi;
  - 41 + 7 = 48

sum2 = fun(8j) + (j/2);
RL ←

- Önce j/2 yapılır.
  - j = 10
  - 10/2 = 5
- sonra fun(8j) yapılır.
  - j = 10
  - 10+4 = 14
  - (14×3) - 1 = 41
- toplama;
  - 41 + 5 = 46

	sum1	sum2
Left to right :	46	48
Right to left :	48	46