학번 : 20171646

이름 : 박태윤

**텍스트이(가) 표시된 사진

자동 생성된 설명**

① a \* b – 1 + c =>

② a \* ( b – 1 ) / c mod d =>

③ ( a – b ) / c & ( d \* e / a – 3 ) =>

④ -a or c = d and e =>

⑤ a > b xor c or d <= 17 =>

⑥ -a + b =>



① a \* b – 1 + c =>

② a \* ( b – 1 ) / c mod d =>

③ ( a – b ) / c & ( d \* e / a – 3 ) =>

④ -a or c = d and e =>

⑤ a > b xor c or d <= 17 =>

⑥ -a + b =>

텍스트이(가) 표시된 사진

자동 생성된 설명

①

left to right라면 (i / 2)부터 실행이 되어 5를 나타낸다. fun(&i)을 실행하는데, i는 현재 10값을 가지고 있으므로 \*k += 4에서 14를 가지게 되며 3 \* (\*k) - 1에서는 41을 리턴하게 된다. 따라서 sum1은 46을 가진다.

sum2에서 fun(&j)를 실행하는데 j역시 처음에 10을 가지고 있으므로 41을 리턴하며 포인터를 가지고 주소를 참조하여 연산을 진행하였기에 j변수 자체가 14값을 가지게 된다. 이후 (j / 2)에서는 (14 / 2)를 하기 때문에 7을 나타내며 41 + 7을 해 sum2는 48을 가지게 된다.

②

right to left라면 sum1에 대해 처음에 fun(&i)부터 실행이 되고 이 때 i의 값이 10이므로 함수는 41을 리턴하고 i는 14값을 가지게 된다. 이후 (i / 2)를 실행하고 이는 7을 나타내게 된다. 따라서 7 + 41을 최종적으로 하게 되면 sum1에는 48값이 들어가게 된다.

sum2에 대해서는 (j / 2)부터 실행해 이는 5를 나타내게 되고 이후 fun(&j)를 실행하여 함수는 41을 리턴한다. 따라서 최종적으로 41 + 5를 실행해 sum2에는 46값이 들어가게 된다.

텍스트이(가) 표시된 사진

자동 생성된 설명

①

x = x + fun(&x)에 대해 x가 현재 3을 나타내고 있고 이후 fun(&x)을 실행하게 된다. 이 함수는 4를 리턴하기 때문에 x = x + fun(&x)는 3 + 4를 하여 x는 7을 나타내게 된다.

②

fun(&x)를 먼저 실행하게 되고 \*i += 5에 의해 x값이 3에서 8로 바뀌게 된다. 이후 이 함수는 4를 리턴하므로 x = x + fun(&x)는 8 + 4를 하게 되어 x는 12를 나타내게 된다.

텍스트이(가) 표시된 사진

자동 생성된 설명



(편의상 두 출력에 대해 1, 2번으로 번호를 부여하겠다.)

1번 출력에 대해서는 a를 10으로 assign한 뒤 b = a + fun()를 실행한다. fun()은 10값을 가지는 a를 받아 a = a + 10을 한 뒤 a를 리턴한다. 우선 순위가 ‘+’보다 fun()을 호출하는 것이 더 높으므로 a의 값을 20으로 먼저 바꾼 후 fun()가 20을 리턴하므로 b는 20 + 20의 값을 가져 40을 출력하게 된다.

2번 출력에 대해서도 역시 fun()을 먼저 호출하여 a의 값을 20으로 바꾸고 20을 리턴하기 때문에 b는 20 + 20의 값을 가져 40을 출력하게 된다.