```
endbr64
                                   push rbp
                                   mov rbp rsp
mov [ rbp - 0x10 ] 0x2a
                                   mov [ rbp - 0x10 ] 0x2a
mov edx [ rbp - 0x10 ]
                                   mov edx [ rbp - 0x10 ]
mov eax [ rbp - 0x10 ]
                                   mov eax [ rbp - 0x10 ]
                                    imul eax edx
                                    sub eax 0x5
                                   mov [ rbp - 0xc ] eax
mov [ rbp - 0xc ] eax
mov edx [ rbp - 0x10 ]
                                   mov edx [ rbp - 0x10 ]
mov eax [ rbp - 0xc ]
                                   mov eax [ rbp - 0xc ]
                                   and eax edx
                                   mov [ rbp - 0x8 ] eax
mov [ rbp - 0x8 ] eax
mov eax [ rbp - 0x8 ]
                                   mov eax [ rbp - 0x8 ]
                                   add eax 0x1
mov [ rbp - 0x4 ] eax
                                   mov [ rbp - 0x4 ] eax
                                   mov eax 0x0
                                   pop rbp
                                    ret
```

endbr64

push rbp

mov rbp rsp

imul eax edx

sub eax 0x5

and eax edx

add eax 0x1

mov eax 0x0

pop rbp

ret

```
VAR = (VAR + NUM);
  int main(){
  int x0 , x1 , x2 , x3 ;
  x0 = 42:
  x1 = ((x0 * x0) - 5);
  x2 = (x0 & x1);
  x3 = (x2 + 1);
  return 0;}
```

VAR = ((VAR * VAR) - NUM);

VAR = (VAR & VAR):

VAR = NUM;