|  |  |
| --- | --- |
| R = X ^ N | -20 N – 16 |
| N | -20 N – 12 |
| X | -20 N – 8 |
| Return Address N | -20 N – 4 |
| Frame Pointer N | -20 N |
| …… | …… |
| R = X ^ 2 | -56 |
| N | -52 |
| X | -48 |
| Return Address 2 | -44 |
| Frame Pointer 2 | -40 |
| R = X | -36 |
| N | -32 |
| X | -28 |
| Return Address 1 | -24 |
| Frame Pointer 1 | -20 |
| R = 1 | -16 |
| N | -12 |
| X | -8 |
| Frame Pointer 0 | -4 |

Assuming the stack pointer starts at 0 (in the code I used 0x1000) in the memory and the stack is in descending order.

To calculate X ^ N, we need (n + 1) stack frame, one in the main and the others for the recursive functions.

The first frame pointer (0) (in the main) points to -4. Starting from the first frame pointer for the function, the i th frame pointer points to -20 i.

The stack pointer starts at 0. Moved to -4, -16. And then in the k th functions moved in the following order: -20 k, -20 k – 16 ……

After the stack pointer reached -20 N – 16, its getting back in the same order.