I wanna explain my program in historical way.

cal <op> <number1> <number2> <output base>

we take these argument from command line.

char* op = op. *nb1 = nuber1. *nb2 = nuber2.

*ob = output base.

then we turn the nuber into long int,

frist we check if there is a '-' and cinylase the type 'b' 'o' 'd' 'x', ~~mark~~ mark them.

In the turn to decimal method, we implement 'number' - '0' and sum them after ~~product~~ $\times 10^n$; for example - '254'

'2' - '0' = 2. '5' - '0' = 5. '4' - '0' = 4 $2\times10^2 + 5\times10 + 4\times10^0 = 254$

how to turn others to decimal? for example '0' '007' '7' - '0' = 7

$7\times8^0 = 7$., so on and so for, each method like this take O(n).

then we do the operation in the integer. which can be handled straightforward in C, then turn the result to the needed type, for example. $16_d \rightarrow$ binary $\rightarrow$ 16%2 = 0 16/2 = 8; 8%2 = 0; 8/2 = 4%2 = 0 4/2 = 2; 2%2 = 0, 2/2 = 1., so. $16_d \rightarrow 10000_b$. and assemble them into a string. '+/-' + 'output base' + 'digre part'. then ~~the~~ printf %