```
//Answer to question #39395, Programming, C++

#include <stdio.h>

int pw(int,int);

int main() {
    int x,n,ans;

    printf("Enter the numbers: base and power\n");

    scanf("%d %d",&x, &n);// reads 2 values from console

    ans = pw( x, n ); // calls the pw function to work with 2 values

    printf("The ans is %d\n",ans);// prints the answer to the console

    return 0;
}


int pw( int x,int n ){
    int y = 1;

    /* If ( condition1) {
        action1
    } else if ( condition2 ) {
        action2
    } else {
        action3
    }
     construction works like this:
     If condition1 is true, program will execute action1, and skip "esle if" and "else"
part;
     If condition1 is false, it will check condition2, and if it is true, program will
execute action2 ,and and skip "else" part;
     If condition1 is false, and condition2 is false, it will execute action3. Else is
interpreted as "in all other cases"
    */

    if ( n == 0 ) { // if the second number (index) is 0 the function returns 0 to the
main function. So in this case our variable ans = 0
        return 0;
    } else if(n == 1) { // if the index is one ( 2 ^ 1 = 2 ). Explanation: taking the "x"
to power of "3" means x * x * x = x ^ 3; so if we take "x" to the power of 1, it would be
just x, that is why any number taken to the power of 1 equals the initial number
        return x;
    } else if ( n == 2 ) {// if the index is 2, we need to multiply "x" 2 times by
itself, so if we substitute 3, it would be 3 * 3 = 9;
        return( x * x );
    } else {
        y = x * x; // as the loop starts with 3 position, we need to take the base
number, which is x take to the power of 2;

        for ( int i = 3; i <= n; i++ ) { // loop is read like this : we're starting with
number 3 (note, this action is done only one time), checking if "n" is bigger or equals
"i" (which is initially 3), and if it is it executes action inside the loop, which is in
our case y = y * x, and then it exectes i++, which is interpreted as i + 1 (it is called
```

```
"increment"), and then it is starting the loop from the beginning, besides "i = 3", of
course;
            y = y * x; // it can also be written as y *= x; It is the same operation, but
the computer executes it faster;
        }
    return y; // it returns "y" onse "i <= n" condition is false
    }
}
```