Fast Power in C++

```
#include <iostream>
using namespace std;

class FastPower {
public:
    static int fastpower(int a, int b) {
        int res = 1;
        while (b > 0) {
            if (b & 1) {
                res = res * a;
            }
            a = a * a;
            b = b >> 1;
        }
        return res;
    }

    static void main() {
        cout << fastpower(3, 5) << endl;
    }
};

int main() {
    FastPower::main();
    return 0;
}
```

Dry Run Table:

Step	b (binary)	b (decimal)	a	res	Operation	Explanation
0	101	5	3	1		Initial values
1	101	5	3	3	res = res * a	LSB is 1 → multiply res by a
2	10	2	9	3	a = a * a, b >>= 1	Square a → $3^2 = 9$, shift b → b = 2
3	10	2	9	3	(skip multiplication)	LSB is 0 → skip multiplying res
4	1	1	81	3	a = a * a, b >>= 1	a = $9^2 = 81$, b = 1
5	1	1	81	243	res = res * a	LSB is 1 → res = 3 × 81 = 243
6	0	0			Done	Loop ends

✓ Final Output:

```
243
```

243