

S1

→ DOUBLE a = 100000;
DOUBLE b = 100000;
DOUBLE c = a * b;
cout << c; → 1e+10

* cout << fixed << c; → 10000000000.000000

* cout << fixed << setprecision(0) << c;
→ 10000000000

→ INT L = 100000;
INT B = 100000;
* cout << L * B; → 1410065408 (Wrong) X
(limit overflow)

* cout << L * LL * B; → 10000000000 (Right) ✓
converts our calculation to long long

NOTE:

$-10^9 < \text{INT} < 10^9$

$-10^{12} < \text{LONG INT} < 10^{12}$

$-10^{18} < \text{LONG LONG INT} < 10^{18}$

(Long long & long long int is same thing)

→ INT T; // (Test cases)
CIN >> T;

* CIN.ignore(); → To take all 3 inputs in getline
while (T--) { if T=3
STRING S;
getline(CIN, S); (→ To take a string along
cout << S << endl; with spaces.
}

NOTE:

We can take maximum 10^{18} length of integer using long long, but what if we need to take 10^{100} length of input, then we can use STRING. We can store our number in string as it has huge capacity of length.

e.g → `STRING S;`

`CIN >> S;` // 12347129726798217

* `INT LAST-DIGIT = S[S.SIZE() - 1] - '0';`

`cout << LAST-DIGIT;` → Accessing last digit as an integer.