INTRO TO C++ · LECTURE 1 · PRACTICE SHEET

Think before you compile

Predict outputs, hunt bugs, then solve real judge problems. Everything here uses only what we covered: types, operators, casting, if/else, loops and scope.

A · Guess the output | B · Debug | C · Write the code | Solutions at the end

PART A Guess the output

Write down your answer **before** running anything. Each snippet sits inside `main()` with `#include <iostream>` and `using namespace std;`.

A1 Division detective ★

```
int a = 17, b = 5;
cout << a / b << " " << a % b << " "
     << a / b * b + a % b;
```

A2 Plus-plus puzzle ★★

```
int x = 5;
int y = x++ + 10;
int z = ++x * 2;
cout << x << " " << y << " " << z;
```

A3 Where does the cast go? ★★

```
double d = 7 / 2 + 7 / 2.0;
cout << d << " " << (int)d * 2
     << " " << (int)(d * 2);
```

A4 Characters are numbers ★★

```
char c = 'a';
c = c + 2;
cout << c << " " << c - 'a'
     << " " << (char)('Z' - 1);
```

A5 Mystery loop ★★

```
int n = 1234, s = 0;
while (n > 0) {
    s = s * 10 + n % 10;
    n /= 10;
}
cout << s;
```

A6 Which x? ★★

```
int x = 1;
for (int i = 0; i < 3; i++) {
    int x = i * 10;
}
cout << x;
```

A7 Below zero ★★

```
unsigned int u = 0;
u--;
cout << u;
```

A8 Chained again ★★★

```
cout << (3 > 2 > 1);
```

In Python, `3 > 2 > 1` is `True`.

A9 Count the steps ★★

```
int cnt = 0;
for (int i = 1; i <= 100; i += 3)
    cnt++;
cout << cnt;
```

A10 Same or not? ★★★

```
int a = 1000000;
long long p = a * a;
long long q = 1LL * a * a;
cout << (p == q);
```

PART B Debug it

Each program has at least one bug. Find it, explain what goes wrong, and fix it.

B1 The average ★

Should print the average of two integers. Input `3 4` should give `3.5`.

```
int a, b;
cin >> a >> b;
double avg = (a + b) / 2;
cout << avg;
```

B2 Is it zero? ★

Input `0` prints `nonzero`. Why?

```
int x;
cin >> x;
if (x = 0) cout << "zero";
else cout << "nonzero";
```

B3 In range? ★★

Should print `inside` only when 10 ≤ n ≤ 20. It prints `inside` for n = 50 too.

```
int n;
cin >> n;
if (10 <= n <= 20) cout << "inside";
else cout << "outside";
```

B4 Counting digits ★★

Should print how many digits n has (n ≥ 0). Works for 4521, fails for one input.

```
int n, digits = 0;
cin >> n;
while (n > 0) {
    n /= 10;
    digits++;
}
cout << digits;
```

B5 Sum of 1 to n ($n \le 10^5$) · there are **three** bugs ★★★

```
int n;
cin >> n;
int sum;
for (int i = 1; i <= n; i++);
    sum += i;
cout << sum;
```

PART C Write the code

Two warm-ups to run on your own machine, then six real judge problems, roughly in order of difficulty. Read the constraints first: they tell you the type to use.

C1 Clock time ★

Read a number of seconds s (0 ≤ s < 86400) and print it as `h:m:s`. For example, `3725` prints `1:2:5`.

> Only `/` and `%` needed. No if/else, no loops.

C2 Leap year ★★

Print YES if year y is a leap year: divisible by 4, except years divisible by 100 that aren't divisible by 400. Test with 2024, 1900, 2000.

> Can you write it as a single condition with `&&` and `||`?

C3 Bear and Big Brother ★

codeforces.com/problemset/problem/791/A

Limak's weight triples each year and Bob's doubles. After how many years is Limak strictly heavier?

> A `while` loop that simulates year by year.

C4 Soldier and Bananas ★★

codeforces.com/problemset/problem/546/A

The i-th banana costs i·k dollars. He has n dollars and wants w bananas. How much must he borrow?

> Loop, or remember the sum 1 + 2 + ... + w. What if he doesn't need to borrow?

C5 Divisibility Problem ★★

codeforces.com/problemset/problem/1328/A

Fewest +1 moves to make a divisible by b. Up to 10^4 tests, a, b ≤ 10^9.

> A loop adding 1 can take 10^9 steps per test. Find a formula using `%`.

C6 Candies and Two Sisters ★★

codeforces.com/problemset/problem/1335/A

Split n candies so Alice gets strictly more than Betty, both > 0. Count the ways. n ≤ $2 \cdot 10^9$.

> List the splits for n = 7 and n = 8 by hand. Spot the pattern.

C7 Odd Divisor ★★★

codeforces.com/problemset/problem/1475/A

Does n have an odd divisor greater than 1? n ≤ 10^{14}, up to 10^4 tests.

> Which numbers have *no* odd divisor? Check the type of n.

C8 Even Odds ★★★

codeforces.com/problemset/problem/318/A

Write 1...n as all the odds, then all the evens. Print the k-th number. n up to 10^{12}.

> You can't build the list. How many odd numbers are there in 1...n?

ANSWER KEY

Solutions

Try first, peek later. All outputs were checked with g++ 13 and every program passes the problem's sample tests.

PART A Outputs

A1	`3 2 17`	17 / 5 truncates to 3, the remainder is 2, and (a/b)·b + a%b always gives back a.
A2	`7 15 14`	x++ uses 5, then x becomes 6, so y = 15. ++x makes x 7 first, so z = 14.
A3	`6.5 12 13`	7/2 is 3 (int), plus 3.5 is 6.5. (int)d·2 chops first: 6·2 = 12. (int)(d·2) doubles first: 13.
A4	`c 2 Y`	'a' + 2 is 'c'. c − 'a' is a plain number (2). 'Z' − 1 cast to char is 'Y'.
A5	`4321`	Each step moves the last digit of n onto the end of s: the loop reverses the number.
A6	`1`	The x inside the loop is a new variable that dies at the closing }. The outer x is untouched.
A7	`4294967295`	unsigned can't go negative, so 0 − 1 wraps to the top: $2^{32} - 1$.
A8	`0`	(3 > 2) is 1, then 1 > 1 is false. C++ reads it left to right, unlike Python.
A9	`34`	i takes 1, 4, 7, ..., 100. That's (100 − 1) / 3 + 1 = 34 values.
A10	`0`	a·a is computed as int and overflows ($10^{12} > 2.1 \cdot 10^9$); p held −727379968 in our run. q is correct.

PART B Fixes

B1	`(a + b) / 2.0`	(a + b) / 2 is integer division, so 7 / 2 = 3 before it's stored in the double.
B2	`if (x == 0)`	`x = 0` assigns 0, and 0 counts as false, so the else branch always runs. `-Wall` warns about this.
B3	`10 <= n && n <= 20`	(10 <= n) is 0 or 1, and both are ≤ 20, so the condition is always true.
B4	handle `n == 0`	For n = 0 the loop never runs and prints 0, but 0 has one digit. Add `if (n == 0) digits = 1;` or use a do-while.
B5	three fixes	(1) The `;` after the for ends the loop, so `i` is out of scope: compile error. (2) `sum` is uninitialized: start at 0. (3) For n = 10^5 the sum is ≈ $5 \cdot 10^9$: use `long long sum = 0;`.

PART C Solutions

C1 Clock time

```
int s;
cin >> s;
cout << s / 3600 << ":"
     << s % 3600 / 60 << ":"
     << s % 60 << '\n';
```

After removing whole hours (% 3600), whatever is left divided by 60 is the minutes.

C2 Leap year

```
int y;
cin >> y;
if ((y % 4 == 0 && y % 100 != 0)
        || y % 400 == 0)
    cout << "YES\n";
else cout << "NO\n";
```

2024 YES · 1900 NO · 2000 YES.

C3 Bear and Big Brother

```
int a, b, years = 0;
cin >> a >> b;
while (a <= b) {
    a *= 3;
    b *= 2;
    years++;
}
cout << years << '\n';
```

`<=`, not `<`: equal weights don't count. Try the sample 1 1.

C4 Soldier and Bananas

```
long long k, n, w;
cin >> k >> n >> w;
long long cost = k * w * (w + 1) / 2;
if (cost > n) cout << cost - n << '\n';
else cout << 0 << '\n';
```

k(1 + 2 + … + w) = k·w(w+1)/2. The trap is printing a negative number when he has enough money.

C5 Divisibility Problem

```
int t;
cin >> t;
while (t--) {
    int a, b;
    cin >> a >> b;
    cout << (b - a % b) % b << '\n';
}
```

a is a % b past a multiple of b, so it needs b − a%b more. The outer % b makes the answer 0 when a%b is already 0.

C6 Candies and Two Sisters

```
int t;
cin >> t;
while (t--) {
    int n;
    cin >> n;
    cout << (n - 1) / 2 << '\n';
}
```

Betty can get 1, 2, … up to just under half. n = 7: 3 ways, n = 8: 3 ways. $2 \cdot 10^9$ still fits in int.

C7 Odd Divisor

```
int t;
cin >> t;
while (t--) {
    long long n;
    cin >> n;
    while (n % 2 == 0) n /= 2;
    if (n > 1) cout << "YES\n";
    else cout << "NO\n";
}
```

Remove every factor of 2. Anything left above 1 is odd. Only powers of 2 answer NO. At most about 47 halvings, since $10^{14} \approx 2^{47}$.

C8 Even Odds

```
long long n, k;
cin >> n >> k;
long long odds = (n + 1) / 2;
if (k <= odds) cout << 2 * k - 1 << '\n';
else cout << 2 * (k - odds) << '\n';
```

1…n has (n+1)/2 odd numbers. The k-th odd is 2k − 1, and the j-th even is 2j. With $n = 10^{12}$, int would overflow.