

Lab Questions

1. Write a program `harmonic.cpp` that takes input an integer n as input and output the n -th harmonic number H_n .

Note: $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$.

2. Write a program `five_per_line.cpp` that, using one `for` loop and one `if` statement, prints the integers from 1000 to 2000 with five integers per line. Only last line may have less than 5 numbers.

Hint: use the `%` operator.

3. Write a program `hellos.cpp` that takes a positive integer n as input and print "Hello" n times, each on a separate line. You may assume that the input is less than 1000.

Hint: Consider using `i % 10` and `i % 100` to determine whether to use "st", "nd", "rd", or "th" for printing the i -th Hello.

Example: With $n = 4$, the program should output the following.

```
1st Hello
2nd Hello
3rd Hello
4th Hello
```

4. *Guessing game:* Generate a random number between 1 and 100. Ask the user to guess the number. Provide feedback (too high, too low) and continue until the user guesses correctly. Use a loop for repetition.
5. A couple beginning a family decides to keep having children until they have at least one of either sex. Assume that the probability of having a boy or girl is $1/2$. Write `happy_family.cpp` to find expected number of children by simulation.
6. Write a program `star_square.cpp` that take input n and use nested `for` loops to produce the following n -by- n square pattern ($n = 5$ in the example below):

```
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

7. Write a program `star_triangle.cpp` that takes an input n and use nested `for` loops to produce the following output ($n = 5$ in the example below):

```
*
* *
* * *
* * * *
* * * * *
```

8. Write a program `triangle.cpp` that takes an input n and prints an n -by- n triangular pattern like the one below ($n = 6$ in the example below):

```
* . . . . .
* * . . . .
* * * . . .
* * * * . .
* * * * * .
* * * * * *
```

9. In the lecture, we saw the an approach to find binary representation of a number n by repeatedly dividing n by 2 and printing the remainder. The method finds the binary digits from right to left.

Here we will implement a different approach which finds binary digits of n from left to right.

1. Let v and k be two integers such that $v = 2^k$ and $v \leq n$. That is, v is the highest power of 2 less than or equal to n .
2. Repeat k times:
 - (a) If v is less than or equal to n , then print 1 and subtract v from n .
 - (b) Otherwise, print 0.
 - (c) Divide v by 2 (integer division).

Write a program `binary.cpp` that takes an integer n as input and prints its binary representation using the method described above.

Hint: k in step 1 can be computed either by using a `while` loop (as discussed in lecture) or by using the `log2()` function from the `cmath` library.

10. *Ramanujan's taxi*: Ramanujan was an Indian mathematician who became famous for his intuition for numbers. When the English mathematician G. H. Hardy came to visit him in the hospital one day, Hardy remarked that the number of his taxi was 1729, a rather dull number. To which Ramanujan replied, "*No, Hardy! No, Hardy! It is a very interesting number. It is the smallest number expressible as the sum of two cubes in two different ways.*" i.e. $1729 = 1^3 + 12^3 = 9^3 + 10^3$.

Write a program `ramanujan.cpp` that takes an integer n as input and prints all integers less than or equal to n that can be expressed as the sum of two cubes in two different ways - find distinct positive integers a , b , c , and d such that $a^3 + b^3 = c^3 + d^3$. Use four nested `for` loops.

Now, the license plate 87,539,319 seems like a rather dull number. Determine why it's not.

C++ common mistakes

- *Gotcha 1.* Will the following code fragment compile? If so, what will it do?

```
int a = 10, b = 18;
if (a = b) std::cout << "equal";
else      std::cout << "not equal";
```

Solution: It uses the assignment operator `=` instead of the equality operator `==` in the conditional. This code fragment will set the variable `a` to `18` and the result of this statement is an integer, which the C++ compiler convert to `bool` value `true`. So, it prints “equal” without an error.

- *Gotcha 2.* What does the following code fragment do?

```
bool a = false;
if (a = true) std::cout << "yes";
else        std::cout << "no";
```

Solution: it prints “yes”. Note that the conditional uses `=` instead of `==`. This means that `a` is assigned the value `true`. As a result, the conditional expression evaluates to `true`. For this reason, it is much better style to use `if (a)` or `if (!a)` when testing `bool` values.

- *Gotcha 3.* What does the following code fragment do?

```
int a = 17, x = 5, y = 12;
if (x > y);
{
    a = 13;
    x = 23;
}
std::cout << a;
```

Solution: Always prints `13` since there is a spurious semicolon after the `if` statement. Thus, the assignment statement `a = 13;` will be executed even though `(x <= y)`. It is legal (but uncommon) to have a block that does not belong to a conditional statement, loop, or method.

- *Gotcha 4.* What does the following code fragment do?

```
for (int x = 0; x < 100; x += 0.5) {
    std::cout << x << std::endl;
}
```

Solution: It goes into an infinite loop printing `0`. The compound assignment statement `x += 0.5` is equivalent to `x = (int)(x + 0.5)`.