

Integer to a Roman numeral

To solve this problem of converting an integer to a Roman numeral, you'll want to simulate the greedy strategy of repeatedly subtracting the largest possible Roman value from num.

[Key Insight:] Create a list of values and their Roman numeral equivalents, sorted from largest to smallest, including the subtractive forms:

| | | | | | | | | | | | | | |
|---------|------|-----|-----|-----|-----|----|----|----|----|----|---|----|---|
| Value: | 1000 | 900 | 500 | 400 | 100 | 90 | 50 | 40 | 10 | 9 | 5 | 4 | 1 |
| Symbol: | M | CM | D | CD | C | XC | L | XL | X | IX | V | IV | I |

[Strategy (Greedy Algorithm):]

1. Start with the largest Roman value.
2. While $\text{num} \geq \text{value}$, subtract value from num and append its corresponding symbol to the result.
3. Move to the next smaller Roman value.
4. Repeat until $\text{num} = 0$.

[Example: num = 1994:]

step-by-step using the table:

• 1000 \rightarrow subtract once \rightarrow "M" \rightarrow num = 994

• 900 \rightarrow subtract once \rightarrow "CM" \rightarrow num = 94

• 90 \rightarrow subtract once \rightarrow "XC" \rightarrow num = 4

• 4 \rightarrow subtract once \rightarrow "IV" \rightarrow num = 0

final output: "MCMXCIV"

[Tip:] This approach avoids all complicated digit parsing (units/tens/etc). You don't need to split num into decimal places - the table handles it.

[Table implementation:] Start by building two arrays:

`int values[] = {...};`

`string symbols[] = {...};`

Then iterate through them with a loop using the greedy logic.