

JĘZYK PROGRAMOWANIA C++

LICZBY W SYSTEMIE RZYMSKIM

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

System rzymski zapisywania liczb wykorzystuje cyfry pochodzenia etruskiego, które Rzymianie przejęli i zmodyfikowali około 500 roku p.n.e. Jest to system addytywny i nadaje się do wygodnego zapisywania liczb, jest jednak niewygodny w prowadzeniu nawet prostych działań arytmetycznych.

W systemie rzymskim używa się 7 liter do zapisu wybranych wartości nominalnych: I (1), V (5), X (10), L (50), C (100), D (500) i M (1000). Aby utworzyć liczbę, trzeba zestawić odpowiednie symbole (zaczynając od litery oznaczającej największy nominal a kończąc na literze oznaczającej nominal najmniejszy) sumujące się do zadanej wartości. Na przykład: 187 to CLXXXVII, czyli C+L+X+X+X+V+I+I (100+50+10+10+10+5+1+1 = 187).

Jeżeli składnik liczby którą zapisujemy jest wielokrotnością wartości nominalnej, wtedy zapisywany jest z użyciem kilku następujących po sobie takich samych symboli, z zachowaniem zasady, by nie pisać czterech tych identycznych symboli po sobie. Jedną z zasad, jest umieszczanie oznaczeń I, X i C z lewej strony nominalu wyższego, stąd zestawienia takie jak: IV, IX, XL, XC, CD i CM.

Zadanie.

Napisz program do przekształcania liczb zapisanych zwykłymi arabskimi cyframi na zapis tej liczby w systemie rzymskim. Liczby w zapisie arabskim należy dostarczyć do programu poprzez argumenty wywołania. Każdy argument wywołania programu to napis typu `const char*`, który najpierw należy przekształcić do postaci binarnej za pomocą własnej funkcji:

```
int arab2bin (const char *x);
```

Funkcja ta ma najpierw sprawdzić, czy napis dostarczony jako argument do tej funkcji składa się z cyfr dziesiętnych, czy nie rozpoczyna się od zera i nie jest dłuższy niż 4 znaki. Jeśli te warunki są spełnione, to można ten napis zinterpretować jako zapis liczby z zakresu od 1 do 9999 i wrócić odpowiadającą mu wartość; w przeciwnym przypadku zwracamy 0. Przy zamianie poprawnego napisu na liczbę całkowitą możesz skorzystać z funkcji bibliotecznej `stoi()` zadeklarowanej w `<string>` w C++14.

Liczbę binarną z kolei należy przekształcić na odpowiadający jej zapis w postaci rzymskiej funkcją:

```
std::string bin2rzym (int x);
```

Funkcja ta ma dla zadanej wartości typu `int` zwrócić rzymski zapis tej wartości jako łańcuch znakowy typu `std::string`. W trakcie tej konwersji skorzystaj operatorów konkatencji i tablicowanych wartości liczbowych i odpowiadających im symboli rzymskich.

```
const int B[] = {1000,
                 900, 500, 400, 100,
                 90, 50, 40, 10,
                 9, 5, 4, 1};
const std::string R[] = {"M",
                         "CM", "D", "CD", "C",
                         "XC", "L", "XL", "X",
                         "IX", "V", "IV", "I"};
```

Program powinien dla każdej prawidłowo podanej wartości wypisać na standardowym wyjściu `std::cout` jej wartość w postaci rzymskiej (każdą liczbę wypisz w osobnej linii). Wszelkie komentarze, czy informacje o błędnych argumentach posyłaj na standardowe wyjście dla błędów `std::cerr`.