

## 1 Задача

Дан шаблон слова. Необходимо

- а) вывести номер заданного слова в лексикографическом порядке
- б) по заданному номеру в лексикографическом порядке вывести слово

## 2 Решение

Номер слова  $N$  можно найти по формуле

$$N = k_1 + k_2 a_1 + k_3 a_2 + k_4 a_3 + \dots + k_n a_{n-1},$$

где  $(k_1, k_2, k_3, \dots, k_n)$  – слово,  $(a_1, a_2, a_3, \dots, a_n)$  – шаблон слова.

Отсюда также следует, что найти элементы слова можно при делении номера  $N$  последовательно на  $a_i$  ( $i = 1, 2, 3, \dots, n$ ). Остаток от каждого деления и есть элемент слова.

### 3 Программа

```
#include <iostream>

#define TEMPLATE_SIZE 5

int main()
{
    int word_template[TEMPLATE_SIZE] = {10, 20, 30, 40, 50};
    int word[TEMPLATE_SIZE] = {0, 0, 0, 0, 0};

    int n = 5123; // найдем пять тысяч сто двадцать третье слово (отсчет с нуля)

    for(int i = TEMPLATE_SIZE-1; i >= 0; i--) {
        word[i] = n % word_template[i]; // соблюдаем порядок букв
        n /= word_template[i];
    }

    // напечатаем найденное слово
    for(int i = 0; i < TEMPLATE_SIZE; i++)
        std::cout << word[i] << ' ';
    std::cout << std::endl;

    // найдем номер нового слова {0, 0, 2, 22, 23}
    int new_word[TEMPLATE_SIZE] = {0, 0, 2, 22, 23};
    int m = 0;
    int p = 1;
    for(int i = TEMPLATE_SIZE-1; i >= 0; i--) {
        m += p*new_word[i];
        p *= word_template[i];
    }
    std::cout << m << std::endl;

    return 0;
}
```

### 4 Работа программы