

Задачи для 1 курса (2023-2024 уч. год.)

4. Задачи на битовые операции

Задачи данного раздела решаются с использованием битовых операций. Многие задачи можно решить не используя циклов, т.е. просто последовательностью нескольких операций.

1. Определить позицию самой старшей единицы в битовом представлении данного целого числа.
2. Написать функцию, которая зеркально переворачивает битовое представление целого числа (аналогично перестановке массива в обратном порядке) в пределах до старшей значащей единицы.
3. Написать функцию, которая циклически сдвигает битовое представление целого числа на указанное количество позиций вправо или влево (аналогично циклическому сдвигу массива).
4. Написать функцию, которая переставляет байты целого числа (int) в обратном порядке.
5. Найти первые N целых чисел, у которых младший байт является зеркальным отражением следующего байта.
6. Проверить четность количества единиц в двоичном представлении данного целого числа.
7. В заданном диапазоне найти все числа с симметричной битовой записью в пределах до старшей значащей единицы
8. Для заданного K вывести все числа из заданного диапазона, в двоичной записи которых присутствует ровно K единиц.
9. Инвертировать биты, стоящие в диапазоне позиций от i до j , в данном целом числе.
10. Для двух целых чисел подсчитать количество двоичных разрядов, в которых они оба содержат 1. Например, для чисел 15 и 21 ответ будет 2.
11. Релизовать представление десятичных чисел в целом числе (4 бита на одну десятичную цифру) и реализовать функции печати таких чисел и операцию сложения для них.
12. Реализовать контейнер для четырех чисел типа char на базе одной переменной типа int, т.е. реализовать функции чтения и записи i -го байта целого числа по указанному индексу $i = 0, 1, 2, 3$.
13. Реализовать функцию, которая в битовом представлении целого числа (32 бита) ищет указанную битовую конфигурацию (k бит). Конфигурация задается k младшими битами другого числа. Требуется найти самую старшую позицию совпадения конфигураций.
14. Возвести число в степень N за не более чем $2 \log_2 N$ умножений.
15. Вывести в файл все подмножества множества $\{1, \dots, N\}$ при $N \leq 32$.
16. Вывести в файл все k -элементные подмножества мн-ва $\{1, \dots, N\}$ при $N \leq 32$.
17. Печатный символ задается пиксельной матрицей 8×8 , которая хранится в двух целых числах (8 байт). Реализовать функцию печати строки из N таких символов, задаваемой массивом unsigned int $c[2*N]$ (по два последовательных элемента на символ), считая, что один пиксел печатается при помощи одной обычной буквы, например, Ш.
18. Двухцветная иконка размера 32×32 пиксела задается массивом из 32 целых чисел (каждое число — строка картинки, 1 — черный, 0 — белый). Требуется определить есть ли на данной иконке вертикальные черные линии (столбик из 1, по бокам которого стоят 0).
19. Двухцветная иконка размера 32×32 пиксела задается массивом из 32 целых чисел (каждое число — строка картинки, 1 — черный, 0 — белый). Требуется определить есть ли на данной иконке выделенная рамка — по границе картинки стоят 1, и эта рамка отделена от остальной части нулями.
20. Двухцветная иконка размера 32×32 пиксела задается массивом из 32 целых чисел (каждое число — строка картинки, 1 — черный, 0 — белый). Требуется реализовать “генератор шахматных досок” с размером клетки в k пикселов. Шахматная доска “прижимается” к левой и верхней границам иконки, а правая и нижняя границы могут обрезать ячейки доски (если k не есть степень двойки).
21. Полоски штрих-кода представляются набором битов — черный цвет 1, белый цвет 0. Числа от 0 до 9 кодируются в шести битах как $0xxxx1$, где $xxxx$ - двоичная запись числа. Например код для 5 есть 001011, код для 9 есть 010011 и т.п. (на самом деле не так, но это здесь не важно). В одной целой переменной (32 бита) можно закодировать 5 шифр штрих-кода. Написать функции преобразования массива цифр в представление штрих-кода туда и обратно.
22. Картинка QR-кода занимает матрицу размером в 32×32 пиксела и задается массивом из 32 целых чисел (каждое число — строка картинки). Для позиционирования QR-кода используется специальный фрагмент — черный квадрат со стороной в 2 пиксела, окруженный рамкой белого цвета и далее рамкой черного цвета. На заданной матрице QR-кода требуется обнаружить такой фрагмент (указать где он лежит) или сказать, что его нет.