

Лабораторная работа №7
Работа с системами счисления.

Индивидуальные задания

Лабораторная работа обязана быть одним решением с проектами. Unit тестирование выполнить для задач с звёздочками, подобрать краевые тесты. Используйте функции ручной проверки на ввод из предыдущей лабораторной работы. Для каждого задания вычислить сложность алгоритма. Около каждой строки кода, где сложность алгоритма больше чем $O(1)$ в комментариях указать сложность алгоритма.

Задание 1. Перевести числа. Предусмотреть ввод положительных и отрицательных чисел.

1. Из прямого кода в обратный
2. Из дополнительного кода в обратный
3. Из обратного кода в естественную форму
4. Из естественной формы в прямой код
5. Из прямого кода в дополнительный
6. Из прямого кода в обратный
7. Из дополнительного кода в обратный
8. Из обратного кода в естественную форму
9. Из естественной формы в прямой код
10. Из прямого кода в дополнительный
11. Из естественной формы в дополнительный код
12. Из естественной формы в обратный код
13. Из дополнительного кода в прямой код
14. Из дополнительного кода в естественную форму
15. Из обратного кода в естественную форму

Задание 2. Осуществить сложение чисел. Разработать функции для выполнения операции сложения. Предусмотреть ввод положительных и отрицательных чисел.

1. Найдите сумму двоичных чисел, заданных в естественной форме. Сложение выполните в дополнительном коде. Ответ выразите в прямом коде.
2. Найдите сумму двоичных чисел, заданных в естественной форме. Сложение выполните в дополнительном коде. Ответ выразите в прямом коде.
3. Найдите сумму двоичных чисел, заданных в естественной форме. Сложение выполните в обратном коде. Ответ выразите в дополнительном коде.
4. Найдите сумму двоичных чисел, заданных в естественной форме. Сложение выполните в обратном коде. Ответ выразите в прямом коде.
5. Найдите сумму двоичных чисел, заданных в естественной форме. Сложение выполните в дополнительном коде. Ответ выразите в прямом коде.

6. Найдите сумму двоичных чисел, заданных в естественной форме. Сложение выполните в дополнительном коде. Ответ выразите в прямом коде.
7. Найдите сумму двоичных чисел, заданных в естественной форме. Сложение выполните в обратном коде. Ответ выразите в прямом коде.
8. Найдите сумму двоичных чисел, заданных в естественной форме. Сложение выполните в обратном коде. Ответ выразите в дополнительном коде.
9. Найдите сумму двоичных чисел, заданных в естественной форме. Сложение выполните в дополнительном коде. Ответ выразите в прямом коде.
10. Найдите сумму двоичных чисел, заданных в естественной форме. Сложение выполните в дополнительном коде. Ответ выразите в естественной форме.
11. Найдите сумму двоичных чисел, заданных в естественной форме. Сложение выполните в обратном коде. Ответ выразите в прямом коде.
12. Найдите сумму двоичных чисел, заданных в естественной форме. Сложение выполните в обратном коде. Ответ выразите в естественной форме.
13. Найдите сумму двоичных чисел, заданных в естественной форме. Сложение выполните в дополнительном коде. Ответ выразите в естественной форме.
14. Найдите сумму двоичных чисел, заданных в естественной форме. Сложение выполните в дополнительном коде. Ответ выразите в естественной форме.
15. Найдите сумму двоичных чисел, заданных в естественной форме. Сложение выполните в обратном коде. Ответ выразите в прямом коде.

Задание 3. Пользователь вводит основание системы счисления. Осуществить сложение и вычитание чисел в заданной системе счисления. В другую систему счисления не переводить. В системах счисления больших десятичной использовать буквы по аналогии с шестнадцатеричной системой. Разработать функции для выполнения операции сложения и функции для выполнения операции вычитания. Предусмотреть ввод положительных и отрицательных чисел.

Задание 4*. Перевести число из арабской системы записи чисел в римскую.

Задание 5*.

Пусть $S(x)$ — это сумма цифр числа x , записанного в десятичной системе счисления. Например, $S(5) = 5$, $S(10) = 1$, $S(322) = 7$.

Мы будем называть число x **интересным**, если $S(x + 1) < S(x)$. В каждом тесте вам будет дано одно число n . Ваша задача — посчитать количество чисел x таких, что $1 \leq x \leq n$ и x — интересное.

Входные данные

Первая строка содержит одно целое число t ($1 \leq t \leq 1000$) — количество наборов входных данных.

Далее следуют t строк, i -я строка содержит одно целое число n ($1 \leq n \leq 10^9$) для i -го набора входных данных.

Выходные данные

Выведите t целых чисел, i -е число должно быть ответом на i -й набор входных данных.

входные данные	выходные данные
5	0
1	1
9	1
10	3
34	88005553
880055535	

Задание 6. Необходимо разработать программу, которая проверяет, делится ли введенное пользователем число на заданное простое. Программа не должна содержать операций умножения, деления, вычитания (в том числе взятия остатка от деления). Рекомендуется использовать побитовые операции. Для каждого варианта нужно проверить делимость на 3 простых числа. Проверять можно по отдельности (сначала получить ответ для первого, затем для второго, затем для третьего).

Варианты:

- 1) 3, 37, 89
- 2) 5, 47, 89
- 3) 7, 29, 149
- 4) 3, 61, 131
- 5) 5, 73, 151
- 6) 7, 23, 197
- 7) 3, 79, 151
- 8) 5, 29, 223
- 9) 7, 73, 109
- 10) 3, 47, 197
- 11) 5, 53, 109

12) 7, 79, 107

13) 3, 23, 107

14) 11, 43, 179

15) 11, 31, 113

Пояснение ко второй части. Любое число можно представить в виде $a = bq + r$, $0 \leq r < b$ (b предполагается положительным). Для любого s , взаимно простого с p , $a : p$ эквивалентно $as = bsq + sr : p$. Данное выражение, в свою очередь, эквивалентно $(bs \bmod p)q + sr : p$. Если b взаимно просто с p , то существует такое s , что $bs \bmod p = 1$. Таким образом, $a : p$ эквивалентно $q + sr : p$, где s специально подобрано.

Также мы знаем, что r – это последняя цифра числа a в b -ичной системе счисления, а q – число, составленное из всех его цифр, кроме последней. Поэтому признаки делимости можно формулировать в таком виде: «Число делится на p тогда и только тогда, когда на p делится сумма исходного числа без последней цифры и последней цифры числа, умноженной на s ».

Пример. Число делится на 13 тогда и только тогда, когда на 13 делится сумма числа без последней цифры и последней цифры числа, умноженной на 4.

Доказательство. $10q + r : 13 \leftrightarrow 40q + 4r : 13 \leftrightarrow q + 4r : 13$.

Пример работы признака:

$1521 \rightarrow 152 + 4 \cdot 1 = 156$;

$156 \rightarrow 15 + 4 \cdot 6 = 39$;

$39 \rightarrow 3 + 4 \cdot 9 = 39$;

39 делится на 13, значит, 1521 делится на 13.

$1687 \rightarrow 168 + 4 \cdot 7 = 196$;

$196 \rightarrow 19 + 4 \cdot 6 = 43$;

$43 \rightarrow 4 + 4 \cdot 3 = 16$;

16 не делится на 13, значит, 1687 не делится на 13. Но обратите внимание, что у 16 и 1687 разные остатки при делении на 13. Данный признак лишь сообщает делимость, но не может сообщить остаток исходного числа, если он не равен нулю.

В компьютере числа представлены в двоичной системе счисления. Зная блочное правило перевода, вы можете работать с числом так, как будто оно записано в 2^k -ичной системе. Ваша задача – найти такие системы счисления, в которых ваш признак будет максимально удобным (быстрым) в использовании. Разумеется, вы можете пользоваться несколькими системами счисления, например, сначала формулой для восьмеричной, а затем формулой для четверичной системы.

Задание 7*.

Брюс недавно получил работу в NEERC (Numeric Expression Engineering & Research Center), где изучают и строят много различных любопытных чисел. Его первым заданием стало исследование двадесятичных чисел. Натуральное число называется **двудесятичным**, если его десятичное представление является суффиксом его двоичного представления; и двоичное и десятичное представление рассматривается без ведущих нулей. Например, $1010 = 10102$, так что **10** двудесятичное число. Числа $101010 = 11111100102$ и $4210 = 1010102$ не являются двудесятичными. Сначала Брюс хочет создать список двудесятичных чисел. Помогите ему найти n -ое наименьшее двудесятичное число.

Входные данные

Одно целое число n ($1 \leq n \leq 10\,000$).

Выходные данные

Вывести одно число - n -ое наименьшее двадесятичное число в десятичном представлении.

Входные данные #1	Выходные данные #1
1	1
Входные данные #2	Выходные данные #2
2	10
Входные данные #3	Выходные данные #3
10	1100

Задача 8*. Вчера на уроке математики Саша узнал о том, что иногда полезно использовать вместо десятичной системы счисления какую-нибудь другую. Однако, учительница не объяснила, почему в системе счисления по основанию b в качестве цифр выбирают числа от 0 до $b - 1$. Немного подумав, Саша понял, что можно выбирать и другие наборы цифр. Например, вместо троичной системы счисления можно рассмотреть систему счисления, где вместо обычных цифр 0, 1, 2 есть цифры 1, 2 и 3. Саша заинтересовался вопросом, а как перевести число n в эту систему счисления? Например, число 7 в этой системе записывается как 21, так как $7 = 2 \cdot 3 + 1$, а число 22 записывается как 211, так как $22 = 2 \cdot 9 + 1 \cdot 3 + 1$.

Входные данные

натуральное число n , $1 \leq n \leq 2 \cdot 10^9$.

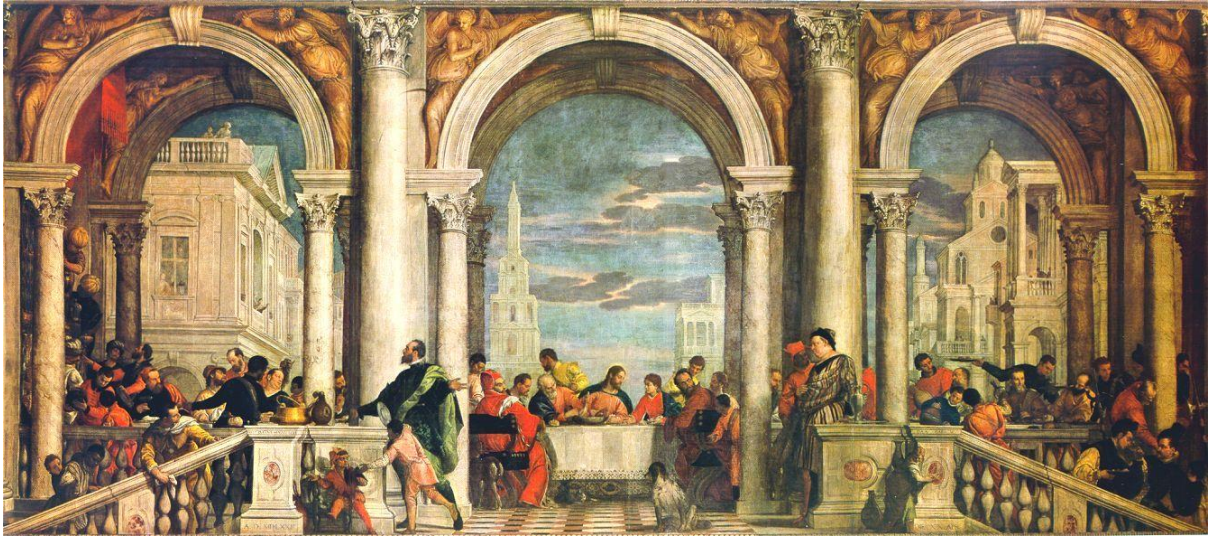
Выходные данные

число n записанное в указанной системе счисления.

Входные данные	Выходные данные
7	21
22	211

Задача 9. Патриций решил устроить праздник и для этого приготовил 240 бочек вина. Однако к нему пробрался недоброжелатель по имени *Пользователь*, который подсыпал яд в одну из бочек. Недоброжелателя тут же поймали, дальнейшая его судьба неизвестна, но ходят слухи, что он проверяет консольный ввод в аду в качестве наказания, однако сейчас не об этом. Про яд известно, что человек, который его выпил, умирает **в течение 24 часов**. До праздника осталось два дня, то есть 48 часов. У патриция есть пять рабов, которыми он готов пожертвовать, чтобы узнать, в какой именно бочке яд. Вы близкий друг Патриция и совершенно не хотите стать одним из тех рабов, что будут проверять вино на

наличие яда. Подойдите к заданию творчески и найдите способ определения отравленной бочки с ядом.



Для решения данной задачи необходимо прочесть тему лабораторной работы и продемонстрировать, что ваш алгоритм решения возвращает номер отравленной бочки, которую отравил пользователь в самом начале.

Тема к защите лабораторной работы: системы счисления, ошибки сегментации указателей, сложность алгоритмов, рекурсия и весь пройденный материал к моменту защиты лабораторной работы.