1. АЛГОРИТМЫ

§1. Неформальное определение алгоритма и примеры алгоритмов

Неформальное определение алгоритма. К числу основных понятий информатики и всей математики относится понятие алгоритма. Дадим неформальное определение этого понятия.

Определение (*неформальное определение алгоритма*)**:** Под алгоритмом решения некоторого типа задач понимают точное описание (правило осуществления) выполняемого шаг за шагом процесса, который завершается через конечное число шагов и приводит к решению любой задачи данного типа.

Для того, чтобы решать с помощью вычислительной машины любую задачу какого-либо класса, нужно заранее разработать полный алгоритм решения задач этого класса. Только после этого машина может быть использована для решения данного класса задач. В случае, если этот алгоритм содержит пробные шаги, разработчик должен выработать систематический метод, дающий возможность машине в случае надобности выполнить эти шаги и принять результат только одной из произведенных проб в качестве требуемого.

Решето Эратосфена. *Простым* называется отличное от единицы натуральное число, которое делится нацело только само на себя и на единицу. Рассмотрим алгоритм нахождения всех простых чисел, меньших заданного числа N. Этот алгоритм, идея которого заключается в вычеркивании всех чисел, меньших N, кратных какому-либо числу, был найден около 250-го года до нашей эры Эратосфеном. Рассмотрим его на примере нахождения всех простых чисел, меньших числа 50.

❑ Сначала выпишем последовательно все натуральные числа от 1 до 49 включительно:

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	

Шаг первый. Вычеркнем 1, так как это – не простое число.

Шаг второй. Оставим первое невычеркнутое число (таковым будет число 2), но вычеркнем каждое второе число после него. Получаем:

~~1~~	2	3	~~4~~	5	~~6~~	7	~~8~~	9	~~10~~
11	~~12~~	13	~~14~~	15	~~16~~	17	~~18~~	19	~~20~~
21	~~22~~	23	~~24~~	25	~~26~~	27	~~28~~	29	~~30~~
31	~~32~~	33	~~34~~	35	~~36~~	37	~~38~~	39	~~40~~
41	~~42~~	43	~~44~~	45	~~46~~	47	~~48~~	49	

Шаг третий. Оставим первое невычеркнутое после 2 число, большее, чем 2 (таковым будет 3), но вычеркнем каждое третье число после него. Получаем:

~~1~~	2	3	~~4~~	5	~~6~~	7	~~8~~	~~9~~	~~10~~
11	~~12~~	13	~~14~~	~~15~~	~~16~~	17	~~18~~	19	~~20~~
~~21~~	~~22~~	23	~~24~~	25	~~26~~	~~27~~	~~28~~	29	~~30~~
31	~~32~~	~~33~~	~~34~~	35	~~36~~	37	~~38~~	~~39~~	~~40~~
41	~~42~~	43	~~44~~	~~45~~	~~46~~	47	~~48~~	49	

Шаг четвертый. Оставим первое невычеркнутое число, большее, чем число 3 (это 5), но вычеркнем каждое пятое число после него. Получаем:

~~1~~	2	3	~~4~~	5	~~6~~	7	~~8~~	~~9~~	~~10~~
11	~~12~~	13	~~14~~	~~15~~	~~16~~	17	~~18~~	19	~~20~~
~~21~~	~~22~~	23	~~24~~	~~25~~	~~26~~	~~27~~	~~28~~	29	~~30~~
31	~~32~~	~~33~~	~~34~~	~~35~~	~~36~~	37	~~38~~	~~39~~	~~40~~
41	~~42~~	43	~~44~~	~~45~~	~~46~~	47	~~48~~	49	

Шаг пятый. Оставим первое невычеркнутое число, большее, чем 5 (это 7), но вычеркнем каждое седьмое число после него. Получаем:

~~1~~	2	3	~~4~~	5	~~6~~	7	~~8~~	~~9~~	~~10~~
11	~~12~~	13	~~14~~	~~15~~	~~16~~	17	~~18~~	19	~~20~~
~~21~~	~~22~~	23	~~24~~	~~25~~	~~26~~	~~27~~	~~28~~	29	~~30~~
31	~~32~~	~~33~~	~~34~~	~~35~~	~~36~~	37	~~38~~	~~39~~	~~40~~
41	~~42~~	43	~~44~~	~~45~~	~~46~~	47	~~48~~	~~49~~	

Шаг шестой. Оставим первое невычеркнутое число, большее, чем 7 (число 11), но вычеркнем каждое одиннадцатое число после него. Мы обнаружим, что на этом шаге из нашего списка не вычеркнуто ни одно новое число. Обнаружив это, выполнение алгоритма завершаем и заявляем, что числа, оставшиеся невычеркнутыми – это все простые числа, меньшие, чем $N=50$. ❑

Вычеркивая каждое второе число после 2, каждое третье число после 3, каждое пятое число после 5 и так далее, мы тем самым вычеркивали числа, кратные 2, 3, 5 и так далее, то есть кратные хотя бы одному простому числу. Иными словами, мы вычеркивали все составные числа. Невычеркнутыми остаются только числа, каждое из которых не является кратным ни одному числу, предшествующему ему, то есть простые числа. Описанный алгоритм часто называют *методом решета Эратосфена*.

Если нужно найти все простые числа, меньшие числа N, то при использовании метода решета Эратосфена процесс заканчивается после того, как оказываются вычеркнутыми все числа, кратные наибольшему натуральному числу, не превосходящему $\sqrt{N}$. В нашем примере таким числом является 7, и процесс в самом деле завершился после того, как были вычеркнуты все числа, кратные 7. Действительно, каждое составное число, меньшее числа N, должно среди своих делителей иметь по крайней мере один простой делитель, не больший, чем $\sqrt{N}$.

Алгоритм разложения данного натурального числа на простые множители. Рассмотрим метод нахождения всех простых делителей данного числа, состоящий в делении этого числа на последовательно взятые простые числа. Проиллюстрируем применение этого метода на примере нахождения всех простых делителей числа 5382.

❑ *Шаг первый*. Проверяем, является ли первое простое число 2 делителем данного числа 5382. Поскольку это число – четное, число 2 является его делителем. Деление 5382 на 2 дает в частном 2691.

Шаг второй. Проверяем, является ли 2 делителем числа 2691. Поскольку число 2691 – нечетное, число 2 не является его делителем.

Шаг третий. Проверяем следующее простое число 3. Сумма цифр числа 2691 равна 18, а 18 делится нацело на 3, значит и 2691 делится нацело на 3. Деление 2691 на 3 дает в частном 897.

Шаг четвертый. Проверяем, является ли 3 делителем 897. Является – по примененному выше признаку делимости. Деление 897 на 3 дает в частном 299.

Шаг пятый. Проверяем, является ли число 3 делителем числа 299. По ранее указанному признаку делимости – не является.

Шаг шестой. Проверяем, является ли следующее простое число 5 делителем числа 299. Не является, поскольку последняя цифра в десятичной записи числа 299 – не 0 и не 5.

Шаг седьмой. Проверяем, является ли следующее простое число 7 делителем числа 299. Записав это число без последней цифры, получаем 29, а удвоенная последняя цифра есть 18. Разность 29 и 18 равна 11, а 11 не делится на 7. Значит и исходное число 299 не делится нацело на 7. Простое число 7 делителем числа 299 не является.

Шаг восьмой. Проверяем, является ли следующее простое число 11 делителем числа 299. На нечетных местах в десятичной записи числа 299 стоят цифры 2 и 9, их сумма равна 11. На единственном четном месте в десятичной записи числа 299 стоит цифра 9. Сумма цифр, стоящих на четных местах, равна 9. Модуль разности суммы цифр числа 299, стоящих на нечетных местах, и суммы цифр, стоящих на четных местах, есть $11-9=2$, а 2 на 11 нацело не делится. Значит и 299 на 11 нацело не делится. Простое число 11 делителем числа 299 не является.

Шаг девятый. Проверяем, является ли следующее простое число 13 делителем числа 299. Количество десятков числа 299 равно 29, а учетверенное количество единиц числа 299 равно $9\cdot 4=36$. Число десятков, сложенное с учетверенным числом единиц, есть $29+36=65$. Число 65 на 13 делится, значит и 299 на 13 делится. Деление 299 на 13 дает в частном 23, являющееся простым числом. На этом выполнение алгоритма завершаем и заявляем, что мы разложили число 5382 на простые множители:

$$\begin{array}{r|r} 5382 & 2 \\ 2691 & 3 \\ 1897 & 3 \\ 299 & 13 \\ 23 & 23 \end{array}$$

$$5382=2\cdot 3^2\cdot 13\cdot 23.$$ ❑

При выполнении рассмотренного выше алгоритма мы использовали признаки делимости, что существенно сокращает описанный процесс, так как при этом исключается необходимость фактически производить некоторые пробные шаги деления. Сформулируем примененные нами признаки делимости.

Критерий (*признак делимости на два*): Натуральное число делится нацело на 2 тогда и только тогда, когда оно – четное. В десятичной записи четного числа последней цифрой является 0, 2, 4, 6 или 8 (их поэтому называют четными цифрами).

Если цифры (то есть знаки, из которых строится десятичная запись натурального числа) рассмотреть поотдельности, то каждая из них оказывается десятичной записью некоторого неотрицательного целого числа, не превышающего девяти. Если указанные целые неотрицательные числа сложить, то получится неотрицательное целое число, называемое *суммой цифр* исходного числа. Аналогичным образом можно говорить о сумме не всех, а только некоторых цифр числа, о данном числе без последней цифры, об удвоенной последней цифре числа и так далее. В общем же, следует помнить, что цифры – это *знаки, а не числа*, поэтому арифметические действия с цифрами проводить нельзя (ибо эти действия определены для *чисел*, а не для *цифр*). Когда же все-таки говорят об арифметических действиях с цифрами, то делают это лишь для сокращения речи, подразумевая действия над числами, запись которых составляется из указываемых цифр.

Критерий (*признак делимости на три*): Натуральное число делится на 3 нацело тогда и только тогда, когда сумма цифр десятичной записи этого числа делится нацело на 3.

Критерий (*признак делимости на пять*): Натуральное число делится на 5 тогда и только тогда, когда последней цифрой в его десятичной записи является 0 или 5.

Критерий (*признак делимости на семь*): Натуральное число делится нацело на 7 тогда и только тогда, когда целое число, десятичная запись которого получается путем вычеркивания из десятичной записи исходного числа последней цифры, после вычитания удвоенной последней цифры делится нацело на 7.

Критерий (*признак делимости на одиннадцать*): Натуральное число делится на 11 нацело тогда и только тогда, когда разность между суммой цифр его десятичной записи, стоящих на нечетных местах, и суммой цифр, стоящих на четных местах, делится нацело на 11.

Критерий (*признак делимости на тринадцать*): Натуральное число делится нацело на 13 тогда и только тогда, когда количество его десятков, сложенное с учетверенным количеством единиц, нацело делится на 13.

Алгоритм Евклида. *Наибольшим общим делителем* двух натуральных чисел называется наибольшее натуральное число, на которое делятся нацело эти оба числа, или, иначе, наибольшее натуральное число, делящее нацело эти числа. Так, например, 6 является наибольшим общим делителем чисел 18 и 24, что кратко обозначается так:

$$6 = \text{НОД}(18, 24).$$

Действительно, никакое другое натуральное число, большее 6, не делит нацело и 18, и 24. Если наибольший общий делитель двух натуральных чисел равен единице (иными словами, если эти числа не имеют других общих делителей, кроме единицы), то они называются *взаимно простыми*.

Для нахождения наибольшего общего делителя двух заданных чисел R_1 и R_2 может быть использована замечательная эффективная процедура, получившая название *алгоритма Евклида*. Ее сущность изложена ниже.

Инструкция C_1: Если $R_1=R_2$, то считать, что НОД$(R_1,R_2)=R_1=R_2$. Завершить работу.
Инструкция C_2: Если $R_1<R_2$, то переименовать переменные так, чтобы было $R_1>R_2$.
Инструкция C_3: Ввести вспомогательный параметр i. Положить значение параметра i равным единице: $i=1$.
Инструкция C_4: Поделить R_i на R_{i+1} с остатком. Остаток обозначить символом R_{i+2}.
Инструкция C_5: Если $R_{i+2}\neq 0$, то увеличить значение вспомогательного параметра i на 1 и перейти к выполнению инструкции C_4.
Инструкция C_6: Если $R_{i+2}=0$, то считать, что НОД$(R_1,R_2)=R_{i+1}$ и на этом завершить работу.

Продолжение следует...

Задачи

1. С помощью решета Эратосфена найдите все простые числа, меньшие числа 110.

2. Разложите на простые множители числа 13356, 16800 и 2835.

3. Используя алгоритм Евклида, найдите: *а*) наибольший общий делитель чисел 6584 и 6624; *б*) наибольший общий делитель чисел 1159 и 1297; *в*) наибольший общий делитель чисел 918 и 1908; *г*) наибольший общий делитель чисел 1134 и 1851; *д*) наибольший общий делитель чисел 504 и 904.

4. Вычислите значение квадратного корня с такой точностью, чтобы полученный Вами результат после проверочного возведения в квадрат и округления до тысячных совпадал бы с подкоренным выражением: *а*) $\sqrt{240}$; *б*) $\sqrt{25{,}83}$; *в*) $\sqrt{207{,}165}$.

5. Запишите предложенные выражения, не используя знак $\sum$:

$$а)\ \sum_{n=1}^{4}\frac{n}{2^{n-1}};\quad б)\ \sum_{n=1}^{5}\frac{(-1)^{n+1}\cdot n}{n^2+1};\quad в)\ \sum_{n=-2}^{0}\left(2^n+\frac{1}{2}\right)^{\frac{n+2}{2}}.$$

6. Запишите предложенные выражения, используя знак $\sum$:

$$а)\ 1+\frac{1}{3}+\frac{1}{5}+\frac{1}{7}+\frac{1}{9}+\frac{1}{11};\quad б)\ x+\frac{x^3}{3}+\frac{x^5}{5}+\frac{x^7}{7}.$$

7. Вычислите значение выражения $\sum_{n=0}^{3} r^n$ при $r=\frac{1}{2}$.

8. Найдите значение выражения $\sum_{n=0}^{5}(1+nd)$ при $d=3$.

9. Вычислите значения следующих выражений:

$$а)\ \sum_{n=0}^{3}(3^n-n);\quad б)\ \sum_{n=1}^{6}(2^n-1);\quad в)\ \sum_{n=-2}^{3}(n^2+1);\quad г)\ \sum_{n=0}^{3}\left(2^{n^3}-n^2-2n\right).$$

10. Вычислите значение выражения 5! двумя способами: используя нерекурсивное и применяя рекурсивное определение факториала.

11. Найдите значения следующих выражений:

$$a)\ \prod_{k=1}^{4} 2^{k-1};\quad б)\ \prod_{x=-1}^{3}\left(2x^2-2^{x+1}+1\right);\quad в)\ \prod_{n=0}^{3}\frac{4^{n-1}}{n!}.$$

12. Найдите значение выражения $\prod\limits_{n=0}^{3} r^{n!}$ при $r=2$.

§2. Формализации понятия алгоритма. Машины Тьюринга

Задачи

13. Входное слово P – непустое слово в алфавите $A=\{0,1\}$. Иначе говоря, слово P – это последовательность двоичных цифр, то есть двоичная запись неотрицательного целого числа в двоичной системе счисления. Требуется после останова машины Тьюринга получить на ленте двоичную запись натурального числа, на единицу большего того, которое было представлено входным словом P.

14. Входное слово P – слово в алфавите $A=\{a,b,c\}$. Машина Тьюринга должна перенести первый символ слова P в его конец, если это слово не является пустым. Пустое входное слово менять не следует.

15. Входное слово P – слово в алфавите $A=\{a,b,c\}$. Если входное слово является пустым, то машина Тьюринга не должна его менять. Если входное слово не является пустым, но первый и последний символы его одинаковы, то это слово машина Тьюринга также должна оставить без изменений. Во всех остальных случаях входное слово должно быть заменено на пустое слово (то есть стерто).

16. Входное слово P – слово в алфавите $A=\{a,b\}$. Машина Тьюринга должна удалить из входного слова его второй символ, если он имеется. Выходное слово должно быть записано на ленте слитно, не разрываясь пустыми ячейками.

17. Входное слово P – слово в алфавите $A=\{a,b,c\}$. Машина Тьюринга должна удалить из входного слова первое вхождение символа a, если оно имеется. Выходное слово должно быть записано на ленте слитно, не разрываясь пустыми ячейками.

18. Входное слово P – слово в алфавите $A=\{a,b,c\}$. Если входное слово не является пустым, то после его первого символа машина Тьюринга должна вставить символ a. Пустое входное слово менять не следует.

19. Входное слово P – слово в алфавите $A=\{a,b,c\}$. Машина Тьюринга должна вставить во входное слово символ a после первого вхождения символа c, если оно имеется.

20. Входное слово P – слово в алфавите $A=\{a,b,c\}$. Машина Тьюринга должна удалить из входного слова все вхождения символа a. Получившееся слово должно быть записано на ленте слитно, не разрываясь пустыми ячейками.

21. Входное слово P – слово в алфавите $A=\{a,b\}$. Машина Тьюринга должна удвоить (скопировать) входное слово, поставив между ним и его копией знак «=».

22. Входное слово P – слово в алфавите $A=\{a,b,c\}$. Машина Тьюринга должна приписать слева к входному слову символ b.

23. Входное слово P – слово в алфавите $A=\{a,b,c\}$. Машина Тьюринга должна приписать справа к входному слову символы b и c.

24. Входное слово P – слово в алфавите $A=\{a,b,c\}$. Машина Тьюринга должна заменить на a каждый второй символ во входном слове.

25. Входное слово P – слово в алфавите $A=\{a,b,c\}$. Если входное слово не является пустым, то машина Тьюринга должна оставить в нем только первый символ. Если же входное слово – пустое, то машина не должна его менять.

26. Входное слово P – слово в алфавите $A=\{a,b,c\}$. Если входное слово не является пустым, то машина Тьюринга должна оставить в нем только последний символ. Если же входное слово – пустое, то машина не должна его менять.

27. Входное слово P – слово в алфавите $A=\{a,b,c\}$. Если $P=ab$, то машина Тьюринга не должна его менять. Если P – какое-либо другое слово, то машина Тьюринга должна его стереть, то есть оставить после останова ленту пустой.

28. Входное слово P – слово в алфавите $A=\{a,b,c\}$. Если во входное слово хотя бы один раз входит символ a, то на ленте машина Тьюринга должна оставить один символ a. Если же в слове P нет ни одного вхождения указанного символа, то машина Тьюринга должна оставить на ленте пустое слово (то есть стереть все символы входного слова, если они имеются).

29. Входное слово P – слово в алфавите $A=\{a,b,c\}$. Если во входное слово не входит символ a, то машина Тьюринга должна заменить в слове P все символы b на c. Если в слове P имеется хотя бы одно вхождение символа a, то в качестве выходного слова машина Тьюринга должна оставить на ленте слово a.

30. Входное слово P есть слово в алфавите $A=\{a,b,0,1\}$. Машина Тьюринга должна определить, является ли входное слово *идентификатором* (то есть непустым словом, начинающимся с буквы, но не с цифры). Роль ответа «да» играет выходное слово a. Роль ответа «нет» играет пустое слово.

31. Входное слово P – слово в алфавите $A=\{a,b,c\}$. Если входное слово имеет четную *длину* (то есть состоит из четного количества букв), то машина Тьюринга должна выдать в качестве выходного слово a. Если же слово P имеет нечетную длину, то машина Тьюринга должна выдать в качестве выходного пустое слово. *Замечание*: пустое входное слово считается имеющим четную длину (нуль букв, а нуль делится нацело на два).

32. Входное слово P – слово в алфавите $A=\{a,b,c\}$, имеющее нечетную длину. Машина Тьюринга должна оставить во входном слове только его средний символ.

33. Входное слово P – слово в алфавите $A=\{a,b,c\}$, имеющее четную длину (пустое слово тоже имеет четную длину). Машина Тьюринга должна оставить во входном слове только левую его половину (пустое слово при этом не изменится).

34. Входное слово P – слово в алфавите $A=\{a,b,c\}$. Машина Тьюринга должна приписать слева к входному слову P его первый символ.

35. Входное слово P – слово в алфавите $A=\{a,b\}$. Машина Тьюринга должна определить, имеется ли в слове P еще хотя бы одно вхождение его первого символа. Если имеется, то на ленте машина должна оставить слово a, в противном случае – пустое слово.

35. Входное слово P – слово в алфавите $A=\{a, b\}$. Машина Тьюринга должна поменять местами первый и последний символы входного слова.

36. Входное слово P – слово в алфавите $A=\{a, b\}$. Машина Тьюринга должна определить, является ли входное слово *палиндромом* (перевертышем, симметричным словом). Если является, то на ленте машина должна оставить слово a, в противном случае – пустое слово. Пустое слово, все однобуквенные слова и некоторые многобуквенные слова вроде $abba$ суть палиндромы – слева направо и справа налево они читаются одинаково. А вот, например, слово $baba$ – не палиндром: если его прочитать справа налево, то получится уже другое слово $abab$.

37. Входное слово P – слово в алфавите $A=\{a, b\}$. Машина Тьюринга должна заменить во входном слове P каждое вхождение символа a на подслово bb.

38. Входное слово P – слово в алфавите $A=\{a, b, c\}$. Машина Тьюринга должна заменить в слове P каждое вхождение подслова ab на c. Выходное слово должно быть записано на ленте слитно, не разрываясь пустыми ячейками.

39. Входное слово P – слово в алфавите $A=\{a, b\}$. Машина Тьюринга должна *удвоить* входное слово, не вставляя между исходным словом и его копией никакого разделителя. Например, поступившему на вход слову abb отвечает выходное слово $abbabb$.

40. Входное слово P – слово в алфавите $A=\{a, b\}$. Машина Тьюринга должна удвоить каждый символ входного слова. Например, входному слову bab отвечает выходное слово $bbaabb$.

41. Входное слово P – слово в алфавите $A=\{a, b\}$. Машина Тьюринга должна *перевернуть* входное слово. Например, поступившему на вход слову abb отвечает выходное слово bba.

42. Входное слово P имеет вид $Q=R$, в котором Q и R – слова в алфавите $A=\{a, b\}$. Машина Тьюринга должна оставить на ленте слово a в том случае, когда слова Q и R являются одинаковыми. В противном случае машина должна оставить на ленте пустое слово.

43. Входное слово P – слово в алфавите $A=\{a, b\}$. Если в этом слове символов a больше, чем символов b, машина Тьюринга должна оставить на ленте слово a. Когда же во входном слове символов a меньше, чем символов b, выходным словом должно быть слово b. Если количества символов a и b во входном слове одинаковы, то выходное слово должно быть пустым.

§3. Нормальные (марковские) алгоритмы

Задачи

44. Входное слово P – слово в алфавите $A=\{a, b, c, d\}$. Во входном слове требуется удалить все вхождения символа c, а затем заменить первое вхождение подслова bb на ddd.

45. Входное слово P – слово в алфавите $A=\{a, b\}$. Требуется преобразовать входное слово так, чтобы в его начале оказались все имеющиеся в нем символы a, а в конце – все символы b (разумеется, точно в тех же количествах, в которых они входят в состав исходного слова).

46. Входное слово P – слово в алфавите $A=\{a, b\}$. Нужно удалить из входного слова его первый символ, если таковой имеется.

47. Входное слово P – непустое слово в алфавите $A=\{0,1,2,3\}$. Иначе говоря, входное слово – это запись неотрицательного целого числа в четверичной системе счисления. Необходимо получить запись того же числа в двоичной системе счисления. Этого можно достичь, осуществив перекодировку входного слова по следующей схеме: каждый символ 0 во входном слове заменяется на 00, 1 – на 01, 2 – на 10 и 3 – на 11. Создайте нормальный (марковский) алгоритм, решающий поставленную задачу.

48. Входное слово P – слово в алфавите $A=\{a,b\}$. Требуется приписать символ a к концу слова P.

49. Входное слово P – слово в алфавите $A=\{a,b\}$. Требуется во входном слове заменить последнее вхождение символа a на двухбуквенное подслово aa. Например, входному слову $bababb$ отвечает выходное слово $babaabb$.

50. Входное слово P – слово в алфавите $A=\{a,b\}$. Если входное слово не является пустым, то нужно перенести в конец его первый символ. Пустое входное слово менять не требуется. Например, входному слову $bbaba$ отвечает выходное слово $babab$.

51. Входное слово P – слово в алфавите $A=\{a,b\}$. Требуется удвоить это слово, то есть приписать к нему слева или справа его копию. Например, входному слову abb отвечает выходное слово $abbabb$.

52. Входное слово – слово в алфавите $A=\{a,b\}$. Это слово имеет четную длину (0, 2, 4 и так далее). Требуется удалить правую половину входного слова. Например, входному слову $bbab$ отвечает выходное слово bb.

53. Входное слово P – слово в алфавите $A=\{f,h,p\}$. Требуется в слове P заменить все вхождения подслова ph на f.

54. Входное слово P – слово в алфавите $A=\{f,h,p\}$. Требуется в слове P заменить на f только первое вхождение подслова ph, если оно имеется.

55. Входное слово P – слово в алфавите $A=\{a,b,c\}$. Требуется приписать слово bac слева к слову P.

56. Входное слово P – слово в алфавите $A=\{a,b,c\}$. Требуется заменить входное слово на пустое слово, то есть удалить из него все символы.

57. Входное слово P – слово в алфавите $A=\{a,b,c\}$. Требуется входное слово, каким бы оно ни было, заменить на выходное слово a.

58. Входное слово P – слово в алфавите $A=\{|\}$. Считая входное слово записью числа в унарной системе счисления, нужно получить остаток от деления этого числа на 2, то есть получить слово, состоящее из одной палочки, если представляемое входным словом число нечетно, или пустое слово, если число, запись которого получена на входе, является четным. *Примечание*: в унарной системе счисления число 0 представляется пустым словом, число 1 – одной палочкой, число 2 – двумя палочками и так далее.

59. Входное слово P – слово в алфавите $A=\{|\}$. Считая входное слово записью положительного числа в унарной системе счисления, нужно уменьшить это число на 1, то есть получить запись в унарной системе счисления числа, на единицу меньшего, чем то, запись которого была представлена входным словом.

60. Входное слово P – слово в алфавите $A=\{|\}$. Считая входное слово записью числа в унарной системе счисления, требуется увеличить это число на 2, то есть получить запись в унарной системе счисления числа, на два большего, чем то, запись которого была представлена входным словом.

61. Входное слово P – слово в алфавите A={0,1,2}. Считая входное слово записью числа в троичной системе счисления, требуется получить остаток от деления этого числа на 2, то есть на выходе получить слово 1, если данное на входе число нечетно, или слово 0, если это число четно. *Примечание*: неотрицательное целое число четно тогда и только тогда, когда в его троичной записи содержится четное количество цифр 1.

62. Входное слово P – слово в алфавите A={a,b,c}. Требуется установить, входит ли символ a в слово P. Ответом должно быть выходное слово a, если входит, или пустое слово, если не входит.

63. Входное слово P – слово в алфавите A={0,1}. Если во входное слово входит больше символов 1, чем символов 0, то в качестве выходного слова требуется получить однобуквенное слово 1. Если во входном слове количества символов 1 и 0 одинаковы, то в качестве ответа должно быть выдано пустое слово. Если же во входном слове символов 0 больше, чем символов 1, то выходным должно быть однобуквенное слово 0.

64. Входное слово P – слово в алфавите A={0,1,2,3}. Требуется преобразовать входное слово так, чтобы в выходном слове сначала шли все четные цифры (0 и 2), а затем – все нечетные (1 и 3).

65. Входное слово P – слово в алфавите A={a,b,c}. Требуется преобразовать входное слово так, чтобы в выходном слове сначала шли все символы a, затем – все символы b, и в конце – все символы c. Разумеется, в выходное слово указанные символы должны входить точно в тех же количествах, что и во входное слово.

66. Входное слово P – слово в алфавите A={a,b,c}. Требуется определить, из скольких различных символов составлено входное слово. Ответ должен быть получен в унарной системе счисления. Например, входному слову *acaac* отвечает выходное слово ||, так как указанное входное слово составлено из двух различных символов a и c.

67. Входное слово P – слово в алфавите A={a,b}. Требуется во входном слове все символы a заменить на b, а все прежние символы b – на символы a.

68. Входное слово P – слово в алфавите A={a,b,c}. Требуется удвоить каждый символ во входном слове. Например, входному слову *bacb* отвечает выходное слово *bbaaccbb*.

69. Входное слово P – слово в алфавите A={a,b}. Требуется приписать слева к входному слову столько палочек, сколько всего символов в него входит. Например, входному слову *babb*, состоящему из четырех символов, отвечает выходное слово ||||*babb*.

70. Входное слово P – слово в алфавите A={a,b}. Известно, что оно имеет четную длину (0, 2, 4 и так далее). Требуется удалить левую половину этого слова. *Примечание*: можно использовать результат решения предыдущей задачи.

71. Входное слово P – слово в алфавите A={a,b}. Требуется приписать справа к входному слову столько палочек, сколько имеется подряд идущих символов a в начале этого слова. Например, входному слову *aababa* отвечает выходное слово *aababa*||.

72. Входное слово P – слово в алфавите A={a,b,c}. Это слово является непустым. Необходимо определить, входит ли первый символ входного слова еще раз в это слово. Ответом является слово a, если указанное вхождение имеется, или пустое слово в противном случае.

73. Входное слово P – слово в алфавите $A=\{a,b\}$. Если входное слово является пустым, то его не следует менять. В противном случае требуется переставить первый и последний символы входного слова.

74. Входное слово P – слово в алфавите $A=\{a,b\}$. Известно, что это слово имеет нечетную длину (1, 3, 5 и так далее). Необходимо удалить из него средний символ.

§4. Машины Поста

Задачи

75. На ленте дана запись неотрицательного целого числа, представленная сплошным непустым массивом меток. Число 0 кодируется массивом длины 1, число 1 – массивом длины 2, число 2 – массивом длины 3 и так далее. Каретка обозревает одну из помеченных ячеек. Требуется после нормального останова машины Поста получить на ленте запись в той же кодировке неотрицательного целого числа, на единицу большего чем то, которое представлял входной массив. Положение каретки на момент останова может быть любым.

76. На ленте дана запись неотрицательного целого числа, представленная сплошным непустым массивом меток. Число 0 кодируется массивом длины 1, число 1 – массивом длины 2, число 2 – массивом длины 3 и так далее. Каретка обозревает одну из ячеек, находящихся левее массива меток. Требуется после нормального останова машины Поста получить на ленте запись в той же кодировке неотрицательного целого числа, на единицу большего чем то, которое представлял входной массив. Положение каретки на момент останова может быть любым.

77. На ленте дана запись неотрицательного целого числа, представленная сплошным непустым массивом меток. Число 0 кодируется массивом длины 1, число 1 – массивом длины 2, число 2 – массивом длины 3 и так далее. Каретка обозревает одну из ячеек, находящихся правее массива меток. Требуется после нормального останова машины Поста получить на ленте запись в той же кодировке неотрицательного целого числа, на единицу большего чем то, которое представлял входной массив. Положение каретки на момент останова может быть любым.

78. На ленте дана запись неотрицательного целого числа, представленная сплошным непустым массивом меток. Число 0 кодируется массивом длины 1, число 1 – массивом длины 2, число 2 – массивом длины 3 и так далее. Каретка обозревает одну из ячеек, находящихся не правее массива меток. Требуется после нормального останова машины Поста получить на ленте запись в той же кодировке неотрицательного целого числа, на единицу большего чем то, которое представлял входной массив. Положение каретки на момент останова может быть любым.

79. На ленте дана запись неотрицательного целого числа, представленная сплошным непустым массивом меток. Число 0 кодируется массивом длины 1, число 1 – массивом длины 2, число 2 – массивом длины 3 и так далее. Каретка обозревает одну из ячеек, находящихся не левее массива меток. Требуется после нормального останова машины Поста получить на ленте запись в той же кодировке неотрицательного целого числа, на единицу большего чем то, которое представлял входной массив. Положение каретки на момент останова может быть любым.

80. На ленте дана запись неотрицательного целого числа, представленная сплошным непустым массивом меток. Число 0 кодируется массивом длины 1, число 1 – массивом длины 2, число 2 – массивом длины 3 и так далее. Начальное положение каретки может оказаться любым. Требуется после нормального останова машины Поста получить на ленте запись в той же кодировке неотрицательного целого числа, на единицу большего чем то, которое представлял входной массив. Положение каретки на момент останова может быть любым.

81. На ленте на расстоянии 1 друг от друга записаны два сплошных непустых массива меток, каждый из которых является закодированной записью неотрицательного целого числа. Любое неотрицательное целое число n кодируется сплошным непустым массивом меток длиной $n+1$. Каретка обозревает одну из помеченных ячеек левого массива. Требуется после нормального останова машины Поста получить на ленте запись в той же кодировке суммы двух указанных чисел. Положение каретки на момент останова может быть любым.

82. На ленте на расстоянии 1 друг от друга записаны не менее двух сплошных непустых массивов меток, каждый из которых является закодированной записью неотрицательного целого числа. Любое неотрицательное целое число n кодируется сплошным непустым массивом меток длиной $n+1$. Каретка обозревает одну из помеченных ячеек самого левого массива. Требуется после нормального останова машины Поста получить на ленте запись в той же кодировке суммы всех указанных чисел. Положение каретки на момент останова может быть любым.

83. На ленте на произвольном расстоянии друг от друга записаны два сплошных непустых массива меток, каждый из которых является закодированной записью неотрицательного целого числа. Любое неотрицательное целое число n кодируется сплошным непустым массивом меток длиной $n+1$. Каретка обозревает одну из помеченных ячеек левого массива. Требуется после нормального останова машины Поста получить на ленте запись в той же кодировке суммы двух указанных чисел. Положение каретки на момент останова может быть любым.

84. На ленте на произвольных расстояниях друг от друга (не больших, чем 3) записаны не менее двух сплошных непустых массивов меток, каждый из которых является закодированной записью неотрицательного целого числа. Любое неотрицательное целое число n кодируется сплошным непустым массивом меток длиной $n+1$. Каретка обозревает одну из помеченных ячеек самого левого массива. Требуется после нормального останова машины Поста получить на ленте запись в той же кодировке суммы всех указанных чисел. Положение каретки на момент останова может быть любым.

85. На ленте дана запись произвольного неотрицательного целого числа n, закодированного сплошным непустым массивом меток, имеющим длину $n+1$. Каретка обозревает одну из помеченных ячеек. Требуется после нормального останова машины Поста получить на ленте запись в той же кодировке результата целочисленного деления данного числа на 2.

86. На ленте дана запись произвольного неотрицательного целого числа n, закодированного сплошным непустым массивом меток, имеющим длину $n+1$. Каретка обозревает одну из помеченных ячеек. Требуется после нормального останова машины Поста получить на ленте запись в той же кодировке результата целочисленного деления данного числа на 3.

87. На ленте дана запись произвольного неотрицательного целого числа n, закодированного сплошным непустым массивом меток, имеющим длину $n+1$. Каретка обозревает одну из помеченных ячеек. Требуется после нормального останова машины Поста получить на ленте запись в той же кодировке результата целочисленного деления данного числа на 4.

88. На ленте дана запись двух неотрицательных целых чисел, каждое из которых закодировано сплошным непустым массивом меток, имеющим длину, на 1 большую, чем кодируемое им число. Расстояние между массивами и начальное положение каретки программисту позволяется определить по своему усмотрению. Требуется составить для машины Поста программу неотрицательного вычитания второго числа из первого. Если неотрицательное вычитание невозможно (вычитаемое больше уменьшаемого), программа не должна приводить к нормальному останову.

89. На ленте дана запись двух неотрицательных целых чисел, каждое из которых закодировано сплошным непустым массивом меток, имеющим длину, на 1 большую, чем кодируемое им число. Расстояние между массивами и начальное положение каретки программисту позволяется определить по своему усмотрению. Требуется составить для машины Поста программу умножения двух данных чисел. Результат должен быть закодирован таким же образом, как и входные данные.

90. На ленте дана запись неотрицательного целого числа, которое закодировано сплошным непустым массивом меток, имеющим длину, на 1 большую, чем кодируемое им число. Начальное положение каретки программисту позволяется определить по своему усмотрению. Требуется составить для машины Поста программу возведения данного числа в квадрат. Результат должен быть закодирован таким же образом, как и входные данные.

91. На ленте дана запись двух неотрицательных целых чисел, каждое из которых закодировано сплошным непустым массивом меток, имеющим длину, на 1 большую, чем кодируемое им число. Расстояние между массивами и начальное положение каретки программисту позволяется определить по своему усмотрению. Требуется составить для машины Поста программу целочисленного деления первого числа на второе. Если деление невозможно (делитель есть нуль), программа не должна приводить к нормальному останову.

2. СИСТЕМЫ СЧИСЛЕНИЯ

§5. Счет. Правила счета

Задачи

92. Формализуйте алгоритм счета в прямом направлении для унарной системы счисления. Иначе говоря, формализуйте алгоритм получения по записи любого неотрицательного целого числа в унарной системе счисления записи в той же системе счисления неотрицательного целого числа, на единицу большего. Например, входному слову ||| отвечает выходное слово ||||.

93. Формализуйте алгоритм счета в обратном направлении для унарной системы счисления. Иначе говоря, формализуйте алгоритм получения по записи любого натурального числа в унарной системе счисления записи в той же системе счисления неотрицательного целого числа, на единицу меньшего. Например, входному слову ||| отвечает выходное слово ||. Пустое входное слово (запись в унарной системе счисления неотрицательного целого числа 0, не являющегося натуральным) алгоритм должен перерабатывать в пустое слово, то есть оставлять без изменений.

94. Формализуйте алгоритм счета в прямом направлении для двоичной системы счисления. Этот алгоритм должен по записи любого неотрицательного целого числа в двоичной системе счисления давать возможность узнать, какова запись в той же системе счисления числа, на единицу большего. Например, входному слову 100 (двоичная запись числа 4) отвечает выходное слово 101 (двоичная запись числа 5), а входному слову 111 (двоичная запись числа 7) – выходное слово 1000 (двоичная запись числа 8).

95. Формализуйте алгоритм счета в обратном направлении для двоичной системы счисления. Этот алгоритм должен по записи любого натурального числа в двоичной системе счисления давать возможность узнать, какова запись в той же системе счисления неотрицательного целого числа, на единицу меньшего. Например, входному слову 1101 (двоичная запись числа 13) отвечает выходное слово 1100 (двоичная запись числа 12), а входному слову 10000 (запись числа 16) – результирующее слово 1111 (двоичная запись числа 15). Входное слово 0 (запись в двоичной системе счисления неотрицательного целого числа 0, не являющегося натуральным) алгоритм должен перерабатывать в слово 0, то есть оставлять без изменений.

96. Формализуйте алгоритм счета в прямом направлении для восьмеричной системы счисления. Этот алгоритм, получая на вход восьмеричную запись неотрицательного целого числа, должен выдавать запись в той же системе счисления числа, на единицу большего. Например, входному слову 56 (восьмеричная запись числа 46) отвечает выходное слово 57 (восьмеричная запись числа 47), а входному слову 37 (восьмеричная запись числа 31) – выходное слово 40 (восьмеричная запись числа 32).

97. Формализуйте алгоритм счета в обратном направлении для восьмеричной системы счисления. Этот алгоритм, получая на вход восьмеричную запись натурального числа, должен выдавать запись в той же системе счисления неотрицательного целого числа, на единицу меньшего. Например, входному слову 132 (восьмеричная запись числа 90) отвечает выходное слово 131 (восьмеричная запись числа 89), а входному слову 110 (восьмеричная запись числа 72) – слово 107 (восьмеричная запись числа 71). Входное слово 0 (запись в восьмеричной системе счисления неотрицательного целого числа 0, не являющегося натуральным) алгоритм должен перерабатывать в слово 0, то есть оставлять без изменений.

98. Формализуйте алгоритм счета в прямом направлении для шестнадцатиричной системы счисления. Иначе говоря, формализуйте алгоритм, преобразующий запись любого неотрицательного целого числа в шестнадцатиричной системе счисления в запись в той же системе счисления числа, на единицу большего. Например, полученному на вход слову DB (шестнадцатиричная запись числа 219) отвечает выходное слово DC (шестнадцатиричная запись числа 220), а входному слову $6F$ (шестнадцатиричная запись числа 111) – выходное слово 70 (шестнадцатиричная запись числа 112).

99. Формализуйте алгоритм счета в обратном направлении для шестнадцатиричной системы счисления. Иначе говоря, формализуйте алгоритм, преобразующий запись любого натурального числа в шестнадцатиричной системе счисления в запись в той же системе счисления неотрицательного целого числа, на единицу меньшего. Например, входному слову DD (шестнадцатиричная запись числа 221) отвечает выходное слово DC (шестнадцатиричная запись числа 220), а полученному на вход слову 60 (шестнадцатиричная запись числа 96) – выходное слово $5F$ (шестнадцатиричная запись числа 95). Входное слово 0 (запись в шестнадцатиричной системе счисления неотрицательного целого числа 0, не являющегося натуральным) должно перерабатываться в слово 0, то есть оставаться неизменным.

100. На вход поступает двоичная запись неотрицательного целого числа. Необходимо определить, является оно четным (выходное слово 0) или нечетным (выходное слово 1). Формализуйте алгоритм решения поставленной задачи.

101. На вход поступает пятиричная запись неотрицательного целого числа. Необходимо найти остаток от деления этого числа на 5 и записать этот остаток в пятиричной системе счисления. Формализуйте алгоритм решения поставленной задачи.

102. На вход поступает семиричная запись неотрицательного целого числа. Необходимо найти результат целочисленного деления этого числа на 49 и записать полученное неполное частное в семиричной системе счисления. Формализуйте алгоритм решения поставленной задачи.

102. На вход поступает троичная запись неотрицательного целого числа. Необходимо найти результат умножения этого числа на 81 и записать полученное произведение в троичной системе счисления. Формализуйте алгоритм решения поставленной задачи.

103. На вход поступает двенадцатиричная запись неотрицательного целого числа. Необходимо узнать, делится это число нацело на 144 (выходное слово 1) или не делится (выходное слово 0). Формализуйте алгоритм решения поставленной задачи.

§6. Перевод записи неотрицательного целого числа из одной системы счисления в другую

Задачи

104. Формализуйте алгоритм перевода записи любого неотрицательного целого числа из двоичной системы счисления в восьмеричную. Например, входному слову 10000100 (двоичная запись числа 132) соответствует выходное слово 204 (восьмеричная запись того же числа). Учтите, что запись, полученная на вход, может содержать стартовые нули.

105. Формализуйте алгоритм перевода записи любого неотрицательного целого числа из восьмеричной системы счисления в двоичную. Например, входному слову 306 (восьмеричная запись числа 198) соответствует выходное слово 11000110 (двоичная запись того же числа). Учтите, что запись, полученная на вход, может содержать стартовые нули, а запись, выдаваемая на выход, стартовых нулей содержать не должна.

106. Требуется перевести запись любого неотрицательного целого числа из двоичной системы счисления в шестнадцатиричную. Исходная запись является корректной, но может содержать стартовые нули. Результат перевода должен, конечно же, также быть корректным и не должен содержать стартовых нулей. Например, входному слову 100000111 (двоичная запись числа 263) соответствует выходное слово 107 (шестнадцатиричная запись того же неотрицательного целого числа). Формализуйте алгоритм, решающий поставленную задачу.

107. На вход поступает запись неотрицательного целого числа в шестнадцатиричной системе счисления, содержащая, быть может, стартовые нули. Необходимо на выходе получить двоичную запись того же числа, не содержащую стартовых нулей. Например, входному слову $15B$ (шестнадцатиричная запись числа 347) отвечает выходное слово 101011011 (двоичная запись того же числа). Выполните формализацию алгоритма, решающего поставленную задачу.

108. Необходимо перевести восьмеричную запись произвольного неотрицательного целого числа, начинающуюся, быть может, со стартовых нулей, в шестнадцатиричную запись того же числа, не содержащую стартовых нулей. Например, входное слово 167 (восьмеричная запись числа 119) должно быть переработано в выходное слово 77 (шестнадцатиричная запись того же числа). Постройте формализацию решающего указанную задачу алгоритма.

109. Формализуйте алгоритм перевода шестнадцатиричной записи произвольного неотрицательного целого числа, содержащей, быть может, стартовые нули, в восьмеричную запись того же числа, не содержащую стартовых нулей. Например, входному слову $67A$ (шестнадцатиричная запись числа 1658) соответствует выходное слово 3172 (восьмеричная запись того же числа).

110. На вход поступает двоичная запись неотрицательного целого числа. Требуется получить десятичную запись того же числа. Исходная запись может содержать стартовые нули, а выходная запись их содержать не должна. Например, входному слову 00011001 (двоичная запись числа 25) отвечает выходное слово 25 (десятичная запись того же числа). Формализуйте алгоритм, решающий поставленную задачу.

111. Требуется переводить десятичные записи неотрицательных целых чисел в соответствующие им двоичные записи. Исходная запись может содержать стартовые нули, но в записи-результате их быть не должно. Формализуйте алгоритм, решающий поставленную задачу. Например, входное слово 00254 (десятичная запись числа 254 со стартовыми нулями) должно перерабатываться в выходное слово 11111110 (двоичная запись того же числа без стартовых нулей).

112. Формализуйте алгоритм перевода десятичной записи неотрицательного целого числа, содержащей, быть может, стартовые нули, в восьмеричную запись того же числа, не содержащую стартовых нулей. Например, входному слову 0157 (десятичная запись числа 157 с одним стартовым нулем) соответствует выходное слово 235 (восьмеричная запись того же числа, не содержащая стартовых нулей).

113. Необходимо перевести восьмеричную запись произвольного неотрицательного целого числа, начинающуюся, быть может, со стартовых нулей, в десятичную запись того же числа, не содержащую стартовых нулей. Например, входное слово 00257 (восьмеричная запись числа 175, содержащая стартовые нули) должно быть переработано в выходное слово 175 (десятичная запись того же числа, не содержащая стартовых нулей). Формализуйте решающий указанную задачу алгоритм.

114. Входное слово представляет собой десятичную запись неотрицательного целого числа, начинающуюся, быть может, со стартовых нулей. Выходное слово должно быть записью того же числа в шестнадцатиричной системе счисления, не содержащей стартовых нулей. Формализуйте алгоритм, решающий поставленную задачу. Например, входному слову 0106 (десятичная запись числа 106 с одним стартовым нулем) отвечает выходное слово 6*A* (шестнадцатиричная запись того же числа без стартовых нулей).

115. Формализуйте алгоритм перевода шестнадцатиричной записи неотрицательного целого числа в его десятичную запись. Исходная запись может начинаться со стартовых нулей, а в записи-результате их быть не должно. Например, входное слово 00089 (шестнадцатиричная запись числа 137, в которой присутствуют стартовые нули) должно перерабатываться в выходное слово 137 (десятичная запись того же числа без стартовых нулей).

§7. Примыкающие к процедурам счисления вычислительные алгоритмы

Задачи

116. Входное слово представляет собой две двоичные записи двух неотрицательных целых чисел, разделенные пробелом. Каждая из них может содержать стартовые нули. Выходное слово должно быть двоичной записью суммы данных чисел, не содержащей стартовых нулей. Формализуйте алгоритм, решающий поставленную задачу. Например, входному слову 00100 101, кодирующему требование найти сумму чисел 4 и 5, соответствует выходное слово 1001 (двоичная запись числа 9).

117. Формализуйте алгоритм вычисления усеченной разности двух неотрицательных целых чисел, представленных своими двоичными записями. Если уменьшаемое больше вычитаемого, усеченная разность равна обычной разности. Если уменьшаемое меньше вычитаемого или равно ему, то усеченная разность равна нулю. Операнды даются входным словом, в котором их двоичные записи (содержащие, быть может, стартовые нули) разделены одним пробелом. Выходная запись стартовых нулей содержать не должна. Например, входному слову 11 00010, кодирующему требование найти усеченную разность чисел 3 и 2, соответствует выходное слово 1 (двоичная запись числа 1).

118. Необходимо найти произведение двух неотрицательных целых чисел. Множители даются входным словом, в котором их двоичные записи (содержащие, быть может, стартовые нули) разделены одним пробелом. Получаемая на выходе двоичная запись искомого произведения не должна содержать стартовых нулей. Например, входному слову 0101 00011, кодирующему требование найти произведение чисел 5 и 3, отвечает выходное слово 1111 (двоичная запись числа 15). Формализуйте решающий указанную задачу алгоритм.

119. Требуется выполнять целочисленное деление одного неотрицательного целого числа на другое. Операнды даются входным словом, в котором их двоичные записи (содержащие, быть может, стартовые нули) разделены одним пробелом. Получаемая на выходе двоичная запись искомого частного стартовых нулей содержать не должна. Если деление невозможно (делитель равен нулю), необходимо сообщить об этом. Формализуйте алгоритм, решающий поставленную задачу. Например, входное слово 0001101 100, кодирующее требование выполнить целочисленное деление числа 13 на число 4, должно перерабатываться в выходное слово 11 (двоичная запись числа 3).

120. Формализуйте алгоритм нахождения остатка от деления одного неотрицательного целого числа на другое. Делимое и делитель представляются входным словом, в котором их двоичные записи (содержащие, быть может, стартовые нули) разделены одним пробелом. Получаемая на выходе двоичная запись остатка от деления стартовых нулей содержать не должна. Если требуемое вычисление осуществить невозможно (делитель равен нулю), необходимо сообщить об этом. Например, входному слову 010101 00010, кодирующему требование найти остаток от деления числа 21 на число 2, соответствует выходное слово 1 (двоичная запись числа 1).

121. Входное слово представляет собой две восьмеричные записи двух неотрицательных целых чисел, разделенные одним пробелом. Каждая из них может содержать стартовые нули. Выходное слово должно быть восьмеричной записью суммы данных чисел, не содержащей стартовых нулей. Формализуйте алгоритм, решающий поставленную задачу. Например, входному слову 0040 075, кодирующему требование найти сумму чисел 32 и 61, соответствует выходное слово 135 (восьмеричная запись числа 93).

122. Формализуйте алгоритм вычисления усеченной разности двух неотрицательных целых чисел, представленных своими восьмеричными записями. Если уменьшаемое больше вычитаемого, усеченная разность равна обычной разности. Если уменьшаемое меньше вычитаемого или равно ему, то усеченная разность равна нулю. Операнды даются входным словом, в котором их восьмеричные записи (содержащие, быть может, стартовые нули) разделены пробелом. Выходная запись стартовых нулей содержать не должна. Например, входному слову 45 00014, кодирующему требование найти усеченную разность чисел 37 и 12, соответствует выходное слово 31 (восьмеричная запись числа 25).

123. Необходимо найти произведение двух неотрицательных целых чисел. Множители даются входным словом, в котором их восьмеричные записи (содержащие, быть может, стартовые нули) разделены одним пробелом. Получаемая на выходе запись вычисленного произведения не должна содержать стартовых нулей. Например, если дано входное слово 020 00012, кодирующее требование найти произведение чисел 16 и 10, то на выходе должно быть получено слово 240 (восьмеричная запись числа 160). Формализуйте решающий эту задачу алгоритм.

124. Требуется выполнять целочисленное деление одного неотрицательного целого числа на другое. Операнды даются входным словом, в котором их восьмеричные записи (содержащие, быть может, стартовые нули) разделены одним пробелом. Получаемая на выходе запись целочисленного частного стартовых нулей содержать не должна. Если деление невозможно (делитель равен нулю), нужно сообщить об этом. Формализуйте алгоритм, решающий поставленную задачу. Например, входное слово 0113 36, кодирующее требование выполнить целочисленное деление числа 75 на число 30, должно перерабатываться в выходное слово 2 (восьмеричная запись числа 2).

125. Формализуйте алгоритм нахождения остатка от деления одного неотрицательного целого числа на другое. Делимое и делитель представляются входным словом, в котором их восьмеричные записи (содержащие, быть может, стартовые нули) разделены одним пробелом. Получаемая на выходе восьмеричная запись остатка от деления стартовых нулей содержать не должна. Если требуемое деление осуществить невозможно (делитель равен нулю), необходимо сообщить об этом. Например, входному слову 70 00024, кодирующему требование найти остаток от деления числа 56 на число 20, соответствует выходное слово 20 (восьмеричная запись числа 16).

126. Входное слово представляет собой две шестнадцатиричные записи двух неотрицательных целых чисел, разделенные пробелом. Каждая из них может содержать стартовые нули. Выходное слово должно быть шестнадцатиричной записью суммы данных чисел, не содержащей стартовых нулей. Формализуйте алгоритм, решающий поставленную задачу. Например, входному слову 0030 03B, кодирующему требование найти сумму чисел 48 и 59, соответствует выходное слово 6B (шестнадцатиричная запись числа 107).

127. Формализуйте алгоритм вычисления усеченной разности двух неотрицательных целых чисел, представленных своими шестнадцатиричными записями. Если уменьшаемое больше вычитаемого, усеченная разность равна обычной разности. Если уменьшаемое меньше вычитаемого или равно ему, то усеченная разность равна нулю. Операнды даются входным словом, в котором их шестнадцатиричные записи (содержащие, быть может, стартовые нули) разделены одним пробелом. Выходная запись стартовых нулей содержать не должна. Например, входному слову 00043 31, кодирующему требование найти усеченную разность чисел 67 и 49, соответствует выходное слово 12 (шестнадцатиричная запись числа 18).

128. Необходимо найти произведение двух неотрицательных целых чисел. Множители даются входным словом, в котором их шестнадцатиричные записи (содержащие, быть может, стартовые нули) разделены одним пробелом. Получаемая на выходе запись произведения не должна содержать стартовых нулей. Например, если на вход получено слово $1F\ 0000C$, кодирующее требование найти произведение чисел 31 и 12, то на выходе должно получиться слово 174 (шестнадцатиричная запись числа 372). Формализуйте решающий эту задачу алгоритм.

129. Требуется выполнять целочисленное деление одного неотрицательного целого числа на другое. Операнды даются входным словом, в котором их шестнадцатиричные записи (содержащие, быть может, стартовые нули) разделены одним пробелом. Получаемая на выходе запись целочисленного частного стартовых нулей содержать не должна. Если деление невозможно (делитель равен нулю), необходимо сообщить об этом. Формализуйте алгоритм, решающий поставленную задачу. Например, входное слово 00030 011, кодирующее требование выполнить целочисленное деление числа 48 на число 17, должно перерабатываться в выходное слово 2 (шестнадцатиричная запись числа 2).

130. Формализуйте алгоритм нахождения остатка от деления одного неотрицательного целого числа на другое. Делимое и делитель представляются входным словом, в котором их шестнадцатиричные записи (содержащие, быть может, стартовые нули) разделены одним пробелом. Получаемая на выходе шестнадцатиричная запись остатка от деления стартовых нулей содержать не должна. Если деление совершить невозможно (делитель равен нулю), необходимо сообщить об этом. Например, входному слову $000F1\ 038$, кодирующему требование найти остаток от деления числа 241 на число 56, соответствует выходное слово 11 (шестнадцатиричная запись числа 17).

131. Входное слово представляет собой две десятичные записи двух неотрицательных целых чисел, разделенные пробелом. Каждая из них может содержать стартовые нули. Выходное слово должно быть десятичной записью суммы данных чисел, не содержащей стартовых нулей. Формализуйте алгоритм, решающий поставленную задачу. Например, входному слову 016 000324 соответствует выходное слово 340.

132. Формализуйте алгоритм вычисления усеченной разности двух неотрицательных целых чисел, представленных своими десятичными записями. Если уменьшаемое больше вычитаемого, усеченная разность равна обычной разности. Если уменьшаемое меньше вычитаемого или равно ему, то усеченная раность равна нулю. Операнды даются входным словом, в котором их десятичные записи (содержащие, быть может, стартовые нули) разделены одним пробелом. Выходная запись стартовых нулей содержать не должна. Например, если на вход получено слово 018 00006, то на выходе должно получиться слово 12.

133. Необходимо найти произведение двух неотрицательных целых чисел. Множители даются входным словом, в котором их десятичные записи (содержащие, быть может, стартовые нули) разделены пробелом. Получаемая на выходе запись произведения не должна содержать стартовых нулей. Например, входному слову 14 00007 отвечает выходное слово 98.

134. Требуется выполнять целочисленное деление одного неотрицательного целого числа на другое. Операнды даются входным словом, в котором их десятичные записи (содержащие, быть может, стартовые нули) разделены одним пробелом. Получаемая на выходе запись целочисленного частного стартовых нулей содержать не должна. Если деление невозможно (делитель равен нулю), необходимо сообщить об этом. Формализуйте алгоритм, решающий поставленную задачу. Например, входное слово 00013 005 должно перерабатываться в выходное слово 2.

135. Формализуйте алгоритм нахождения остатка от деления одного неотрицательного целого числа на другое. Делимое и делитель представляются входным словом, в котором их десятичные записи (содержащие, быть может, стартовые нули) разделены пробелом. Получаемая на выходе десятичная запись остатка от деления стартовых нулей содержать не должна. Если деление выполнить невозможно (делитель равен нулю), нужно сообщить об этом. Например, если на вход получено слово 22 0011, то на выходе должно получиться слово 0.