

Лабораторная работа N 2

Цели и задачи работы: изучение функций ввода-вывода данных, программирования вычисления значения выражения.

Задание к работе:

1. Реализовать линейный вычислительный процесс. Самостоятельно решить задачу в соответствии с индивидуальным вариантом. Реализовать представленные задачи на языках программирования C++, Python, Haskell (только одно задание), C#, Go, Ruby (только одно задание), Java.
2. Реализовать линейный вычислительный процесс любого задания на языке программирования Assembler.
3. Представленные задачи можно реализовать на каждом языке в одной программе с последовательным выполнением.

Методика выполнения работы:

1. Определить типы используемых в программе данных.
2. Описать переменные.
3. Написать функции ввода-вывода.
4. Разработать алгоритм решения задачи по индивидуальному заданию.
5. Написать и отладить программу с вводом-выводом информации
6. Протестировать работу программы на различных исходных данных.
7. Изменить формат вывода, проверить работу программы при другом формате вывода.

Требования к отчету:

Отчет должен содержать титульный лист, задание, текст программы с комментариями, тесты.

Контрольные вопросы по лабораторной преподаватель задает в процессе демонстрации студентом полученных навыков работы в интегрированной среде.

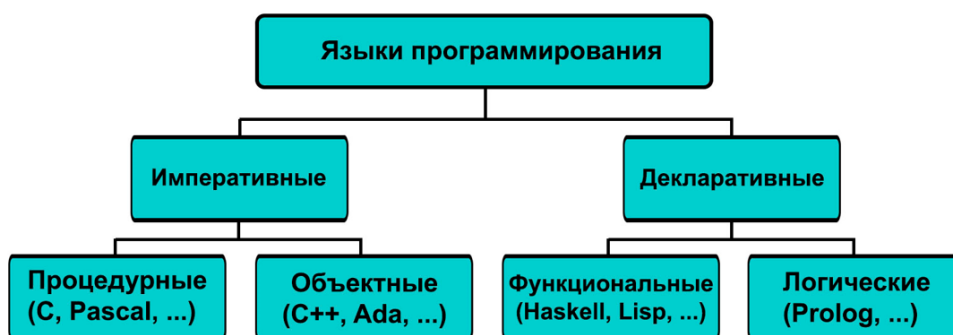
Перечень понятий к защите лабораторной работы 2.

Динамическая типизация, статическая типизация, интерпретатор, компилятор, семантика, рефакторинг, Debug, псевдокод, транслятор, машинный код, сопроцессор, реверс-инжиниринг, дизассемблирование, ассемблерные вставки, дизассемблирование, трансляция, регистры в asm, эзотерический язык, функциональные языки, логические языки, идентификатор, кодировки юникод, ASCII, utf-8, языки низкого уровня, среднего уровня, языки высокого уровня, компилируемые языки, интерпретируемые языки, нативные языки, аллокация.
Императивные и декларативные языки программирования.

Мультипарадигмальный язык, репозиторий, индекс tiobe, высокий уровень безопасности, открытость исходного кода, байт-код, распределенное программирование, автоматическое управление памятью, кроссплатформенность

Апплеты, фреймворк

Знать классификацию ЯП.



Индивидуальные задания:

ЗАДАНИЕ 1.

Все входные и выходные данные в заданиях этой группы являются вещественными числами.

Задание 1 выполняется по вариантам (по модулю 25).

1. Дана сторона квадрата a . Найти квадрат его периметра $P=4 \cdot a$.
2. Дана сторона квадрата a . Найти его площадь $S = a^2$.
3. Даны стороны прямоугольника a и b . Найти его площадь $S = a \cdot b$ и периметр $P = 2 \cdot (a + b)$.
4. Дан диаметр окружности d . Найти ее длину $L = \pi \cdot d$. В качестве значения π использовать 3.14.
5. Дана длина ребра куба a . Найти объем куба $V = a^3$ и площадь его поверхности $S = 6 \cdot a^2$.
6. Даны длины ребер a , b , c прямоугольного параллелепипеда. Найти его объем $V = a \cdot b \cdot c$ и площадь поверхности $S = 2 \cdot (a \cdot b + b \cdot c + a \cdot c)$.
7. Найти длину окружности L и площадь круга S заданного радиуса R : $L=2 \cdot \pi \cdot R$, $S=\pi \cdot R^2$. В качестве значения π использовать 3.14.
8. Даны два числа a и b . Найти их среднее арифметическое: $(a+b)/2$.
9. Даны два неотрицательных числа a и b . Найти их среднее геометрическое, то есть квадратный корень из их произведения.
10. Даны два ненулевых числа. Найти сумму, разность, произведение и частное их квадратов.
11. Даны два ненулевых числа. Найти сумму, разность, произведение и частное их модулей.
12. Даны катеты прямоугольного треугольника a и b . Найти его гипотенузу c и периметр P .
13. Даны два круга с общим центром и радиусами R_1 и R_2 ($R_1 > R_2$).
Найти площади этих кругов S_1 и S_2 , а также площадь S_3 кольца, внешний радиус которого равен R_1 , а внутренний радиус равен R_2 :
 $S_1 = \pi \cdot (R_1)^2$, $S_2 = \pi \cdot (R_2)^2$, $S_3 = S_1 - S_2$.
В качестве значения π использовать 3.14.
14. Дана длина L окружности. Найти ее радиус R и площадь S круга, ограниченного этой окружностью, учитывая, что $L=2 \cdot \pi \cdot R$, $S= \pi \cdot R^2$. В качестве значения π использовать 3,14.
15. Дана площадь S круга. Найти его диаметр D и длину L окружности, ограничивающей этот круг, учитывая, что $L= 2 \cdot \pi \cdot R$, $S= \pi \cdot R^2$. В качестве значения π использовать 3.14.
16. Найти расстояние между двумя точками с заданными координатами x_1 и x_2 на числовой оси: $|x_2 - x_1|$.

17. Даны три точки A, B, C на числовой оси. Найти длины отрезков AC и BC и их сумму.
18. Даны три точки A, B и C на числовой оси. Точка C расположена между точками A и B. Найти произведение длин отрезков AC и BC.
19. Даны координаты двух противоположных вершин прямоугольника: (x_1, y_1) , (x_2, y_2) . Стороны прямоугольника параллельны осям координат. Найти периметр и площадь данного прямоугольника.
20. Поменять местами содержимое переменных A и B и вывести новые значения A и B.
21. Даны переменные A, B, C. Изменить их значения, переместив содержимое A в B, B – в C, C – в A, и вывести новые значения переменных A, B, C.
22. Найти значение функции $y = 3x^6 - 6x^2 - 7$ при данном значении x.
23. Найти значение функции $y = 4(x-3)^6 - 7(x-3)^3 + 2$ при данном значении x.
24. Дано значение угла α в градусах ($0 < \alpha < 360$). Определить значение этого же угла в радианах, учитывая, что $180^\circ = \pi$ радиан. В качестве значения π использовать 3.14.
25. Дано значение угла α в радианах ($0 < \alpha < 2\pi$). Определить значение этого же угла в градусах, учитывая, что $180^\circ = \pi$ радиан. В качестве значения π использовать 3.14.

ЗАДАНИЕ 2

Задание 2 выполняется по вариантам (по модулю 10).

Вариант 1

Римские цифры – цифры, использовавшиеся древними римлянами в их непозиционной системе счисления.

Натуральные числа записываются при помощи повторения этих цифр. При этом, если бóльшая цифра стоит перед меньшей, то они складываются (принцип сложения), если же меньшая стоит перед большей, то меньшая вычитается из большей (принцип вычитания). Последнее правило применяется только во избежание четырёхкратного повторения одной и той же цифры.

Дробей римляне избегали так же упорно, как и больших чисел. В практических задачах, связанных с измерениями, они не использовали дроби, подразделяя единицу измерения обычно на 12 частей, с тем чтобы результат измерения представить в виде составного числа, суммы кратных различных единиц, как это делается сегодня, когда длину выражают в ярдах, футах и дюймах. Английские слова «ounce» (унция) и «inch» (дюйм) происходят от латинского слова [лат. uncia](#) (унция), обозначавшего одну двенадцатую основной единицы длины (таблица 1, 2).

Таблица 1

1	I	лат. unus, unum
5	V	лат. quinque
10	X	лат. decem
50	L	лат. quinquaginta
100	C	лат. centum
500	D	лат. quingenti
1000	M	лат. mille

Таблица 2

Число	Обозначение	Число	Обозначение
1	I	50	L
2	II	60	LX
3	III	70	LXX
4	IV	80	LXXX
5	V	90	XC
6	VI	100	C
7	VII	200	CC
8	VIII	300	CCC
9	IX	400	CD
10	X		

11	XI		
20	XX		
30	XXX		
40	XL		

Основные правила:

если большая цифра стоит перед меньшей, они складываются:

$VI = 5 + 1 = 6$; $XV = 10 + 5 = 15$; $LX = 50 + 10 = 60$; $CL = 100 + 50 = 150$;

если меньшая цифра стоит перед большей (в этом случае она не может повторяться), то меньшая вычитается из большей; вычитаться могут только цифры, обозначающие 1 или степени 10; уменьшаемым может быть только цифра, ближайшая в числовом ряду к вычитаемой:

$IV = 5 - 1 = 4$; $IX = 10 - 1 = 9$; $XL = 50 - 10 = 40$; $XC = 100 - 10 = 90$.

Необходимо написать программу по преобразования натурального числа в римское число.

Пример 1:

Input: LXVI

Output: 66

$I = 1$ $V = 5$ $X = 10$ $L = 50$

Пример 2:

Input: LXXXI

Output: 81

$I = 1$ $X = 10$ $L = 50$

Вариант 2

Вам дано целое число n — количество команд в турнире со странными правилами:

- Если текущее количество команд **четное**, каждая команда объединяется с другой командой. Всего $n / 2$ сыграно матчей, и $n / 2$ команды проходят в следующий раунд.
- Если текущее количество команд **нечетное**, одна команда случайным образом продвигается в турнире, а остальные распределяются по парам. Всего $(n - 1) / 2$ сыграно матчей, и $(n - 1) / 2 + 1$ команды проходят в следующий раунд.

Возвращает количество матчей, сыгранных в турнире, пока не будет определен победитель.

Пример 1:

Input: 7

Output: 6

Объяснение: Детали турнира:

- 1-й раунд: Команды = 7, Матчи = 3, Проходят 4 команды.
- 2-й раунд: Команды = 4, Матчи = 2, и проходят 2 команды.
- 3-й раунд: Команды = 2, Матчи = 1, и 1 команда объявляется победителем.

Общее количество совпадений = $3 + 2 + 1 = 6$.

Пример 2:

Input: 14

Output: 13

Объяснение: Детали турнира:

- 1-й раунд: Команды = 14, Матчи = 7, 7 команд проходят дальше.
- 2-й раунд: Команды = 7, Матчи = 3, Проходят 4 команды.
- 3-й раунд: Команды = 4, Матчи = 2, и проходят 2 команды.
- 4-й раунд: Команды = 2, Матчи = 1, и 1 команда объявляется победителем.

Общее количество совпадений = $7 + 3 + 2 + 1 = 13$.

Вариант 3

Есть n колец, и каждое кольцо белого, синего или красного цвета. Кольца распределены по десяти стержням, помеченным от 0 до 9.

Вам дана строка `rings` длины $2n$, которая описывает n кольца, надетые на стержни. Каждые два символа `rings` образуют пару цвет-позиция, которая используется для описания каждого кольца, где:

- Первый символ пары обозначает цвет кольца (, ,). $i^{th}i^{th}$ 'R''G''B'
- Второй символ пары обозначает стержень, на который надето кольцо (до). $i^{th}i^{th}$ '0''9'

Например, "W3B2R1" описывает $n == 3$ кольца: белое кольцо надевается на стержень с номером 3, синее кольцо надевается на стержень с номером 2 и красное кольцо надевается на стержень с номером 1.

Верните количество стержней, на которых есть кольца всех трех цветов .

Пример 1:

Input: R0R6B0W6W0W6B9

Output: 1

Пример 2:

Input: R0W0B0W9W0R0B0

Output: 1

Пример 3:

Input: B4

Output: 0

Вариант 4

Панграмма (с [греч.](#) – «все буквы»), или **разнобуквица**, – короткий [текст](#), использующий все или почти все [буквы алфавита](#), по возможности не повторяя их. Учитывая строку `sentence`, содержащую только строчные буквы латинского алфавита, вернуть `true`, если это *панграмма*, иначе `false`.

Пример 1:

Input: thequickbrownfoxjumpsoverthelazydog

Output: true

Пример 2:

Input: whatdoesthefoxsay

Output: false

Пример 3:

Input: the five boxing wizards jump quickly

Output: true

Пример 4:

Input: the five boxing wizards jump quickly

Output: true

Пример 5:

Input: how vexingly quick daft zebras jump

Output: true

Пример 6:

Input: the jay, pig, fox, zebra and my wolves quack

Output: true

Пример 7:

Input: how vexingly quick daft zebras jump

Output: true

Ограничения:





















- длина строки ограничена 1000
- строка состоит из строчных английских букв.




Вариант 5

Азбука Мórзе, «Морзя́нка», Код Мórзе – способ знакового кодирования, представление букв алфавита, цифр, знаков препинания и других символов последовательностью сигналов: длинных (тире) и коротких (точек). За единицу времени принимается длительность одной точки. Длительность тире равна трём точкам. Пауза между элементами одного знака – одна точка, между знаками в слове – 3 точки, между словами – 7 точек. Назван в честь американского изобретателя и художника Сэмюэля Морзе.

Буквенные коды (собственно «азбука») были добавлены коллегой Морзе, Альфредом Вейлем – факт, который Морзе впоследствии всячески отрицал (а заодно приписывал себе изобретение телеграфа как такового). Вейлем же, возможно, была придумана и цифровая часть кода. А в 1848 году код Вейля/Морзе был усовершенствован немцем Фридрихом Герке (англ.). Код, усовершенствованный Герке, используется до настоящего времени.

Таблица соответствий:

Русский символ	Латинский символ	Код Морзе
<u>А</u>	 <u>A</u>	· –
<u>Б</u>	 <u>B</u>	– · · ·
<u>В</u>	 <u>W</u>	· – –
<u>Г</u>	 <u>G</u>	– – ·
<u>Д</u>	 <u>D</u>	– · ·
<u>Е</u> (также и <u>Ё</u>)	 <u>E</u>	·
<u>Ж</u>	 <u>V</u>	· · · –
<u>З</u>	 <u>Z</u>	– – · ·
<u>И</u>	 <u>I</u>	· ·
<u>Й</u>	 <u>J</u>	· – – –
<u>К</u>	 <u>K</u>	– · –
<u>Л</u>	 <u>L</u>	· – · ·
<u>М</u>	 <u>M</u>	– –
<u>Н</u>	 <u>N</u>	– ·
<u>О</u>	 <u>O</u>	– – –
<u>П</u>	 <u>P</u>	· – – ·
<u>Р</u>	 <u>R</u>	· – ·
<u>С</u>	 <u>S</u>	· · ·
<u>Т</u>	 <u>T</u>	–
<u>У</u>	 <u>U</u>	· · –
<u>Ф</u>	 <u>F</u>	· · – ·
<u>Х</u>	 <u>H</u>	· · · ·
<u>Ц</u>	 <u>C</u>	– · – ·
<u>Ч</u>	Ö	– – – ·

Русский символ	Латинский символ	Код Морзе
<u>Ш</u>	CH	- - - -
<u>Щ</u>	 Q	- - . -
<u>Ъ</u> ^[8]	Ñ	- - . - -
<u>Ы</u>	 Y	- . - -
<u>Ь</u> (также и <u>б</u>)	 X	- . . -
<u>Э</u>	É	. . - . .
<u>Ю</u>	Ü	. . - -
<u>Я</u>	Ä	. - . -

Дан массив строк, где каждое слово может быть записано как с позиций азбуки Морзе. Определить количество уникальных слов на азбуке Морзе при условии, что буквы можно переставлять. Принять длину слова в строке не более 11 символов. Количество слов не ограничено.

Пример 1:

Input: ign igg ezn smg

Output: 2

Примечание: Строка содержит только 2 вида кодов: "--....." "--.....".

Пример 2:

Input: b

Output: 1

Ограничения:

- `1 <= words.length <= 100`
- `1 <= words[i].length <= 12`
- `words[i]` состоит из строчных английских букв.

Вариант 6

Учитывая массив `nums` целых чисел, вернуть, сколько из них содержат **нечетное количество** цифр.

Пример 1:

Input: 1224,34455,1,8,17596

Output: 2

Пример 2:

Input: 4545,911,102,1035

Output: 2

Вариант 7

Дан массив пар, содержащих начальный и конечный пункты пребывания. Вернуть итоговый конечный пункт самой длинной цепочки следования.

Пример 1:

Input: ["Новосибирск", "Дубай"], ["Новосибирск", "Искитим"], ["Искитим", "Линево"]

Output: Линево

Пример 2:

Input: ["Лондон", "Стамбул"], ["Новосибирск", "Лондон"], ["Стамбул", "Пекин"]

Output: Пекин

Пример 3:

Input: ["Барнаул", "Новосибирск"]

Output: Новосибирск

Пример 4:

Input: ["Искитим", "Искитим"]

Output: обратитесь к специалисту

Вариант 8

Адрес электронной почты (email) – индивидуальный адрес, который присваивается каждому пользователю для получения электронных писем. Email состоит из уникального имени пользователя почты, значка «@», обозначающего, что это email-адрес, и доменного имени.

- Например, в адресе arhipova@corp.nstu.ru, [arhipova](#) – имя пользователя, [corp.nstu.ru](#) – доменное имя.

Ограничения по длине

Имя пользователя может состоять из 6–30 знаков и содержать буквы, цифры и символы.

Специальные символы на практике:

- имя пользователя может содержать буквы латинского алфавита (a–z), цифры (0–9) и точки (.);
- запрещено использовать амперсанд (&), знаки равенства (=) и сложения (+), скобки (<>), запятую (,), символ подчеркивания (_), апостроф ('), дефис (-) и несколько точек подряд;
- имя пользователя может начинаться и заканчиваться любым разрешенным символом, кроме точки (.).

Необходимо вычислить количество уникальных адресов при условии, что точка, оставленная внутри имени пользователя будет проигнорирована, а использование * в имени пользователя игнорирует набор символов после (в пределах имени пользователя).

Например:

- mar.pha@corp.nstu.ru, marpha@corp.nstu.ru и marph.a@corp.nstu.ru в условиях задачи равны.
- mar.pha+science@corp.nstu.ru, marpha+scie.nce@corp.nstu.ru и marph.a+s.c.i.e.n.c.e+@corp.nstu.ru в условиях задачи равны.

Пример 1:

Input: mar.pha+science@corp.nstu.ru, marpha+scie.nce@corp.nstu.ru и marph.a+s.c.i.e.n.c.e+@corp.nstu.ru

Output: 1

Пример 2:

Input: mar.pha+science@co.rp.nstu.ru, marpha+scie.nce@corp.nstu.ru и marph.a+s.c.i.e.n.c.e+@corp.nstu.ru

Output: 2

Вариант 9

У Вас на даче растут цветы, которым требуется постоянный полив. Цветы расположены в ряд и нумеруются от 0 до $n-1$ слева направо. Рядом течет река, в которой вы можете наполнить свою лейку при $x = -1$.

Примечание: каждый цветок нуждается в поливе определенным количество воды.

Алгоритм полива цветов:

- Направление: слева направо.
- Если после полива i -ого цветка не хватает полива для $(i+1)$ цветка, то нужно вернуться и набрать полную лейку.
- Режим экономии: лейки можно наполнять в реке только тогда, когда это требуется.

Для решения задачи необходимо предусмотреть вариант нахождения рядом с рекой в позиции -1 . Для перемещения можно шагать по горизонтальной оси на 1.

Необходимо определить число шагов для полива всех цветов для заданного количества и потребности каждого цветка.

Пример 1:

Input: 3

2, 2, 3, 3, 3

5

Output: 14

Пример 2:

Input: 6

1, 1, 1, 4, 2, 3

4

Output: 30

Пример 3:

Input: 7

7 7 7 7 7 7 7

8

Output: 49

Вариант 10

У Вас есть n банок с водой. Обмену на одну полную банку подлежат m пустых банок.

Необходимо вычислить максимальное число банок, которое пользователь может выпить до положения количества пустых банок, равного одной. А также количество итераций. Отобразить пошаговый процесс (как в примере 1, символы можно изменить в программе).

Пример 1:

Input: 9 3

Output: 13 5

Дано

☺☺☺

☺☺☺

☺☺☺

1 шаг

☹☹☹

☹☹☹

☹☹☹

2 шаг

☺

☺

☺

3 шаг

☹

☹

☹

4 шаг

☺

5 шаг

☹

Пример 2:

Input: 15, 4

Output: 19

Пример 3:

Input: 12, 3

Output: 17

ЗАДАНИЕ 3

Задание на лабораторную работу разбито на несколько вариантов, каждый студент выполняет свой индивидуальный вариант. В каждом варианте требуется выделить из общей задачи более простую подзадачу и написать функцию, которая её решает, после чего применить эту функцию для решения всей задачи. Во всех задачах нельзя использовать ничего из того, что не было ещё изучено (Никаких строк и массивов!). Общая часть для всех заданий: С клавиатуры вводится число n , после чего вводятся n чисел.

Задание 3 выполняется по вариантам (по модулю 25).

Задание 1

1. Необходимо каждое из этих чисел перевернуть и вывести без ведущих нулей.
Пример. Вход: 3 35 20 103 Выход: 53 2 301.
2. Необходимо посчитать количество чётных цифр для каждого числа.
Пример. Вход: 3 12 5 44 Выход: 1 0 2.
3. Определить сумму всех цифр, которые делятся на 3, для всех чисел вместе.
Пример. Вход: 3 92 102 33 Выход: 15.
4. Посчитать количество чисел, у которых сумма цифр больше 10.
Пример. Вход: 4 87 91 4 332 Выход: 1.
5. Для каждого числа определить, является ли оно простым (Простыми называются числа, которые делятся ТОЛЬКО на 1 и на самих себя).
Пример. Вход: 4 17 32 5 9 Выход: 2.
6. Для каждого трёхзначного числа определить является ли оно палиндромом (Читается одинаково с любой стороны, например 101, 222 и т.д.).
Пример. Вход: 5 22 3 101 43 987 Выход: 101 – YES, 987 – NO.
7. Посчитать количество чисел с чётной длиной.
Пример. Вход: 3 123 4444 22 Выход: 2.
8. Определить, сколько чисел из последовательности равны предыдущему числу.
Пример. Вход: 5 12 12 23 23 108 Выход: 2.
9. Посчитать количество чисел, являющихся полными квадратами (их корень является целым числом). Пример. Вход: 4 55 81 32 49. Выход: 2.
10. Для каждого числа из последовательности определить, можно ли поставить внутри числа символ «:» таким образом, чтобы получилась корректная запись времени в формате часы:минуты. Пример. Вход: 3 1244 322 999 Выход: YES YES NO.
11. Вывести все числа, у которых сумма цифр состоит из одного десятичного разряда.
Пример. Вход: 4 19 93 44 29 Выход: 44.
12. Определить количество неубывающих серий в последовательности (неубывающая серия – последовательность подряд идущих чисел, в которой каждое следующее число не меньше предыдущего). Пример. Вход: 5 1 2 3 2 5 Выход: 2.
13. Вывести все числа, которые составлены из цифры только одного вида (например, 1111, 222, 9). Пример. Вход: 3 908 33 72 Выход: 33.
14. Вывести номера чисел из последовательности, у которых сумма цифр меньше их произведения (например, у числа 87 сумма цифр равна 15, а

произведение

56).

Пример. Вход: 3 27 14 99 Выход: 1 3.

15. Все числа из последовательности, которые составлены из цифр, идущих по возрастанию необходимо перевернуть и вывести на экран.
Пример. Вход: 4 87 129 33 45 Выход: 921 54.
16. Для каждого числа из последовательности определить, можно ли поставить внутри числа символ «/» таким образом, чтобы получилась корректная запись даты в формате день:месяц. Пример. Вход: 3 1244 32 199 Выход: NOYESYES.
17. Посчитать сумму всех чисел, у которых на нечётной позиции стоит нечётная цифра (нумерация позиций в числе идёт начиная с 0 справа налево).
Пример. Вход: 4 101 32 21 112 Выход: 144.
18. Определить сумму всех цифр, которые делятся на 3, для всех чисел вместе.
Пример. Вход: 3 92 102 33 Выход: 15.
19. Определить, сколько чисел из последовательности равны предыдущему числу.
Пример. Вход: 5 12 12 23 23 108 Выход: 2.
20. Вывести все числа, у которых сумма цифр состоит из одного десятичного разряда.
Пример. Вход: 4 19 93 44 29 Выход: 44.
21. Вывести номера чисел из последовательности, у которых сумма цифр меньше их произведения (например, у числа 87 сумма цифр равна 15, а произведение 56).
Пример. Вход: 3 27 14 99 Выход: 1 3.
22. Все числа из последовательности, которые составлены из цифр, идущих по возрастанию необходимо перевернуть и вывести на экран.
Пример. Вход: 4 87 129 33 45 Выход: 921 54.
23. Необходимо каждое из этих чисел перевернуть и вывести без ведущих нулей.
Пример. Вход: 3 35 20 103 Выход: 53 2 301.
24. Необходимо посчитать количество чётных цифр для каждого числа.
Пример. Вход: 3 12 5 44 Выход: 1 0 2.
25. Необходимо посчитать количество нечётных цифр для каждого числа.
Пример. Вход: 3 12 5 444 Выход: 0 0 1.