

Лабораторная работа №2

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

Задание к работе:

1. Реализовать линейный вычислительный процесс. Самостоятельно решить задачи в соответствии с индивидуальным вариантом. Реализовать представленные задачи на языках программирования C++, Kotlin, C#, PHP, Rust, Go (Golang), TypeScript, Swift, JavaScript, Python, Java, Ruby.

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:

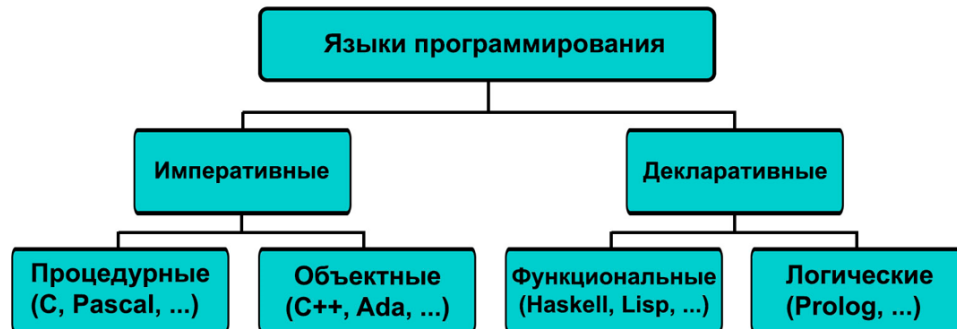
1. отчет должен содержать титульный лист, задание, текст программы с комментариями, тесты;
2. контрольные вопросы по лабораторной преподаватель задает в процессе демонстрации студентом полученных навыков работы в интегрированной среде.

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, 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, утилита, драйвер, 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, сопроцессор, процессор, АЛУ (понятие, структура, операции), основные команды языка

ассемблер, регистр, классификация регистров, регистры общего назначения – структура, указатели команд, регистр флагов, сегментные регистры, сборка проекта на ассемблере, виды ассемблеров, Драйвера, стек, Дистрибутив программы, Парсер, Фреймворк, Кроссплатформенность, АЛУ, Процессор, Кэш.

Проблемы безопасности языков программирования. Трансляция, виртуализация. Знать классификацию ЯП.



Вопросы и понятия с докладов:

Assembler

1. В чём отличие регистра от ячейки памяти?
2. Чем отличаются регистры al/ah/ax/eax/raх?
3. Что такое машинное слово? Длины машинных слов.
4. Основные регистры процессора.
5. Существует ли управление памятью в Assembler?
6. Как можно создать переменную в Assembler?
7. Един ли Assembler для всех архитектур?
8. Связь Assembler и яп C.
9. Для чего был необходим математический сопроцессор во времена 16-битных процессоров Intel?
10. Современный ассемблер: возможно ли одной программой захватить весь компьютер?

Python

1. Python язык компилируемый или интерпретируемый?
2. Особенности динамической типизации Python.
3. Основные типы данных Python и их особенности.
4. Что такое CPython? Что позволяет CPython?
5. Чем занимается Python Software Foundation?
6. Чем вдохновился Гвидо ван Россум при создании своего языка?
7. Перечислите особенности ЯП Python.
8. Области применения Python.
9. С какими языками Python может интегрировать?
10. Как Python повышает читаемость кода?
11. Какие недостатки у Python?
12. В чём проявляется кроссплатформенность Python?

Java

1. Какая компания создала язык Java?
2. Что такое JVM?
3. Что такое многопоточное программирование?
4. Как работает компиляция в Java?
5. Что такое JIT?
6. Что такое Java SE?
7. Что такое javac? Что он делает?
8. Основные преимущества языка Java?
9. Что такое JDK и для чего он нужен?
10. В чём заключается платформенная нейтральность ЯП Java.
11. Процесс компиляции и исполнения программ в Java.
12. Что такое байт-код Java?
13. Базовые типы данных Java.
14. В чём разница между byte и boolean (количество информации, принимаемые значения, занимаемое место)? Отличие int и Integer
15. Что такое Null и в чём его проблема?
16. Как работает сборщик мусора в Java?

JS

1. Для чего был создан JavaScript?
2. Основные ограничения песочницы JS.
3. В чём схожесть Java и JavaScript в методике исполнения кода?
4. Какой движок JS является наиболее распространённым?
5. Различия переменных var, let, const.
6. Динамическая типизация JS. В чём особенности преобразования типов в JS?
7. JavaScript интерпретируемый или компилируемый?
8. Лучшие среды разработки на JS?
9. Этапы интерпретирования JavaScript?
10. Какими были прошлые названия JavaScript?
11. Какие браузеры в настоящее время поддерживают JS?
12. Сферы применения JavaScript
13. Что не умеет делать JavaScript в браузере?

Go:

1. Как реализована кроссплатформенность Go?
2. Что такое каналы в Go?
3. Что такое пакеты в go?
4. Что такое goroutine? Для чего они нужны?
5. Основные типы данных в Go.
6. Управление памятью в Go.
7. Во что компилируется Go?
8. Каковы ограничения на размер стека goroutine в Go, и как эти ограничения зависят от архитектуры системы?
9. Что такое срезы (slices) в Go и как они работают?
10. Что такое нулевое значение (zero value) в Go?
11. Основные преимущества языка Go?
12. Почему язык Go не подходит для Объектно-ориентированного программирования?

C#

1. Устройство архитектуры .NET
2. Принцип работы сборщика мусора
3. Что такое Инкапсуляция, Абстракция, Наследование, Полиморфизм в ООП
4. Что такое Управляемый код, что такое CLR и принцип работы CLR.
5. Базовые типы данных C#.
6. Особенности типизации в C#.
7. Сходства и различия C# и Java.
8. Обработка исключений в C#.
9. Распределение типов данных в памяти.
10. Как работает метод `Console.ReadLine()`?

PHP

1. Что такое JIT? Как он работает?
2. Какая типизация используется в PHP?
3. Как работают переменные и их области видимости в PHP?
4. В чем различие между получением данных из формы методом GET и POST?
5. Как работают функции и анонимные функции в PHP? В чем их особенности?
6. Как работают "магические" методы в PHP (такие как `__get`, `__set`, `__call`, и т.д.)? В каких случаях их следует использовать?
7. В чем различие между операторами сравнения `==` и `===`?
8. Как работает `foreach` и в чем его отличие от `for`?

Kotlin:

1. Принцип платформенной независимости Kotlin.
2. Процесс компиляции и исполнения программ в Kotlin.
3. Отличие типов данных от Java.
4. Объявление переменных в Kotlin. Изменяемые и неизменяемые переменные. Null-safe переменные.
5. В чём отличие метода от функции в Kotlin?
6. Возможно ли использовать в одном проекте Java и Kotlin? Ответ обосновать.

Rust:

1. Особенности ООП на Rust.
2. Что такое асинхронная модель программирования? Что обеспечивает её работоспособность в Rust?
3. Как реализуется многопоточные программы на Rust?
4. Что такое Mutex?
5. В чём особенности управления памятью в Rust? Чем это обеспечивает безопасность программ на этом ЯП?
6. Что такое Cargo? Возможности Cargo.

Swift:

1. Объявление переменных. В чём разница между `let` и `var`?
2. Этапы компиляции программ на Swift.
3. Благодаря чему возможна интероперабельность с Objective-C?

4. Основные аспекты безопасности Swift.
5. Области применения Swift.
6. История, сообщества ЯП Swift.
7. Что такое опциональные (Optional) типы?
8. Что такое ARC?

Ruby

1. Кто создал язык программирования Ruby и в каком году это произошло?
2. Какие основные цели преследовал Юкихио "Matz" Мацумото при создании Ruby?
3. Какие языки и парадигмы программирования повлияли на разработку Ruby?
4. Какие ключевые особенности делают Ruby уникальным среди других языков программирования?
5. Как Ruby on Rails повлиял на популярность Ruby в веб-разработке?
6. Какие известные компании и платформы используют Ruby и Ruby on Rails в своих проектах?
7. Какие основные недостатки Ruby по сравнению с другими языками программирования?
8. Какие инструменты и IDE чаще всего используются для разработки на Ruby?
9. Как Ruby поддерживается сообществом и какие ресурсы доступны для разработчиков?
10. В каких сферах, помимо веб-разработки, активно используется Ruby?

Индивидуальные задания

Первое задание (Модуль 16)

1. Дана строка S . Необходимо найти все ее подстроки, которые удовлетворяют шаблону вида $(1)1[0]1$, где:

(1) – может быть либо 1, либо отсутствовать;

$[0]$ – содержит как минимум один 0 и не включает других символов.

Пример: `str="1101gdyY101tf11f1b10001"`

Результат: 1101, 101, 10001

2. Дана строка S , содержащая как буквы, так и цифры. Найдите такую подстроку, состоящую только из цифр, у которой сумма цифр максимальна. Если таких подстрок несколько, можно вывести любую.

Пример: `str="abc1234def56gh789"`

Результат: 789 (сумма $7+8+9=24$)

3. Даны две строки S и T . Определите, можно ли получить строку T циклическим сдвигом символов строки S .

Пример: `S="waterbottle", T="erbottlewat"`

Результат: Yes

4. Дана строка S . Необходимо преобразовать строку в её сжатую форму, заменив последовательности повторяющихся символов на сам символ и количество его повторений. Если сжатая версия оказывается не короче исходной строки, следует вернуть исходную строку.

Пример: `S="aabcccccaaa"`

Результат: "a2b1c5a3"

5. Студенту не хочется выполнять лабораторную работу. Он должен начать выполнять ее немедленно, чтобы уложиться в срок.

Задание - напечатать строку. За раз он может выполнить одну из следующих операций:

- добавить символ в конце строки;
- добавить копию текущей строки в конец (можно применить единожды).

Помогите студенту найти минимальное количество операций, необходимых для ввода заданной строки.

Пример: `S='dbadbaa'`

Результат: 5 = $d \rightarrow db \rightarrow dba \rightarrow dbadba \rightarrow a$

6. Учитель дал студенту задание: требуется найти N-й член ряда. N-й член – это разность N и ближайшего простого числа к N. Помогите ему найти N-й член для данного N.

Пример: N = 12

Результат: 1

Пример: N = 13

Результат: 0

7. Для заданного натурального числа N и позиции P (отсчитанной с конца, где P = 1 означает последнюю цифру) найдите цифру, которая находится на P-ой позиции в десятичной записи числа N.

Пример: N = 12345, P = 2

Результат: 4

8. Вам задано число N и вывод, напечатанный по определенному паттерну. Необходимо реализовать данный паттерн.

Пример: N= 2

Результат: B

B A

Пример: N= 4

Результат: D

D C

D C B

D C B A

9. Вам дана строка S, содержащая только буквы. Необходимо определить количество гласных и согласных букв, после чего вывести сообщение в соответствии со следующими условиями:

- Если количество гласных > количество согласных, то вывести - “Да”.

- Если количество гласных < количество согласных, то вывести - “Нет”.

- Если количество гласных = количество согласных, то вывести - “Одинаково”.

Пример: S='I like avtf'

Результат: Нет

Пример: S='give me a toy'

Результат: Да

10. Для заданного натурального числа N найдите количество конечных нулей в факториале числа N (то есть в записи $N!$).

Пример: $N=25$

Результат: 6

11. Существует шахматная доска $N \times M$, по которой ходит фигура. Изначально фигура стоит на одной из клеток доски. Задана строка S с командами, задающими направление ходов (L, R, D, U). Один ход = перемещение на 1 клетку. Необходимо определить, возможно ли совершить все заданные команды, не выходя за пределы шахматной доски. Если это возможно, вывести клетку, с которой фигура начинает движение.

Пример: $N = 1, M = 1, S = "L"$

Результат: No (шахматная доска состоит из одной клетки, при перемещении влево фигура выйдет за ее пределы)

Пример: $N = 3, M = 2, s = "DDR"$

Результат: (1,1)

12. Дана бинарная строка и число K . Найдите максимальную длину непрерывной последовательности единиц, которую можно получить, заменив не более K нулей на единицы.

Пример: $S="110100110", K = 2$

Результат: 5

13. Поперек реки находятся N валунов. Стая птиц хочет перебраться с одного берега на другой. Каждая i -ая птица может за раз преодолеть расстояние $birds[i]$. Например, птица за раз может перелететь 2 валуна. Соответственно, ее путь будет состоять из 2, 4, 6 ... валунов. Необходимо найти количество валунов, на которых не появится ни одной птицы за все время пересечения реки.

Пример: $N = 2, stones = 6, birds[] = \{3, 2\}$

Результат: 2 (1 и 5 валуны)

Пример: $N = 2, stones = 6, birds[] = \{5,1\}$

Результат: 0 (вторая птица посетит каждый камень)

14. В частном секторе расположены N подряд стоящих вышек, обеспечивающих сотовую связь. Вам дается строка S длины N , где $S[i] = '1'$ означает, что вышка работает и обеспечивает кооператив связью, $S[i] = '0'$ означает отсутствие связи. Вышка имеет показатель X , который означает, сколько кооперативов слева и справа от себя она может обеспечивать связью. Вам нужно выяснить, могут ли все кооперативы пользоваться сотовой связью.

Пример: $N = 4$, $X = 1$, $S = "1001"$

Результат: Yes

Пример: $N = 4$, $X = 0$, $S = "1001"$

Результат: No

15. Даны две строки: S и T . Найдите минимальную по длине подстроку S , которая содержит все символы T (с учётом кратности).

Пример: $S = "ADOBECODEBANC"$, $T = "ABC"$

Результат: "BANC"

16. Путешественник застрял на острове на S дней в понедельник. На этом острове есть только один магазин, который не работает по воскресеньям. В день он может купить максимально N единиц еды, при этом каждый день ему необходимо съесть M единиц. Необходимо найти минимальное количество дней, когда ему нужно покупать еду в магазине чтобы выжить.

Пример: $S = 5$, $N = 6$, $M = 2$

Результат: 2 (необходимо купить 6 единиц еды в первый день, затем пополнить запасы на 4 день)

Пример: $S = 4$, $N = 5$, $M = 8$

Результат: -1 (выжить невозможно, в день он потребляет еды больше, чем может купить)

Второе задание (Модуль 10)

1. 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:

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

2. 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

3. 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

4. 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

5. 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 = $7 + 3 + 2 + 1 = 13$.

6. 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		лат. <i>unus, unum</i>
5		лат. <i>quinque</i>
10		лат. <i>decem</i>
50		лат. <i>quinginta</i>
100		лат. <i>centum</i>
500		лат. <i>quingenti</i>
1000		лат. <i>mille</i>

Таблица 2

Число	Обозначение	Число	Обозначение
-------	-------------	-------	-------------

			ние
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		

Необходимо написать программу по преобразования римского числа в натуральное 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

7. 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 (, ,).`ithith'R''G''B'`

Второй символ пары обозначает **стержень**, на который **надето** кольцо (до).`ithith'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

8. 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 количеством 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

9. 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.

10. Азбука Мórзе, «Морзья́нка», 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>	– . – .

Русский символ	Латинский символ	Код Морзе
<u>Ч</u>	Ö	- - - .
<u>Ш</u>	CH	- - - -
<u>Щ</u>	 Q	- - . -
<u>Ъ</u> ^[8]	Ñ	- - . - -
<u>Ы</u>	 Y	- . - -
<u>Ь</u> (также и <u>ь</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: "--...-." и "--...-.-".

Пример 2:

Input: b

Output: 1

Ограничения:

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

Третье задание (Модуль 28)

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.

1. Для заданного натурального числа N вычислите сумму его цифр и произведение его цифр, затем найдите модуль разности между этими значениями.

Пример: 3 324 423 500
Результат: 9 24 15 ($2+3+4=9$; $2*3*4=24$; $|9-24|=15$)
9 24 15 ($4+2+3=9$; $4*2*3=24$; $|9-24|=15$)
5 0 5 ($5+0+0=5$; $5*0*0=0$; $|5-0|=5$)

2. Для каждого трёхзначного числа определить является ли оно палиндромом (Читается одинаково, с любой стороны, например 101, 222 и т.д.).

Пример: 5 22 3 101 43 987
Результат: 101 – YES, 987 – NO.

3. Все числа из последовательности, которые составлены из цифр, идущих по возрастанию необходимо перевернуть и вывести на экран.

Пример: 4 87 129 33 45
Результат: 921 54.

4. Необходимо каждое из этих чисел перевернуть и вывести без ведущих нулей.

Пример: 3 35 20 103
Результат: 53 2 301

5. Необходимо посчитать количество чётных цифр для каждого числа.

Пример: 3 12 5 44
Результат: 1 0 2

6. Необходимо каждое из этих чисел перевернуть и вывести без ведущих нулей.

Пример: 3 35 20 103

Результат: 53 2 301

7. Необходимо посчитать количество чётных цифр для каждого числа.

Пример: 3 12 5 44

Результат: 1 0 2

8. Определить сумму всех цифр, которые делятся на 3, для всех чисел вместе.

Пример: 3 92 102 33

Результат: 15

9. Для каждого числа из последовательности определить, можно ли поставить внутри числа символ «:» таким образом, чтобы получилась корректная запись времени в формате часы:минуты.

Пример: 3 1244 322 999

Результат: YES YES NO

10. Вывести все числа, у которых сумма цифр состоит из одного десятичного разряда.

Пример: 4 19 93 44 29

Результат: 44

11. Определить количество неубывающих серий в последовательности (неубывающая серия – последовательность подряд идущих чисел, в которой каждое следующее число не меньше предыдущего).

Пример: 5 1 2 3 2 5

Результат: 2

12. Вывести все числа, которые составлены из цифры только одного вида (например, 1111, 222, 9).

Пример: 3 908 33 72

Результат: 33.

13. Посчитать количество чисел, у которых сумма цифр больше 10.

Пример: 4 87 91 4 332

Результат: 1

14. Для каждого числа определить, является ли оно простым.

Пример: 4 17 32 5 9

Результат: 2

15. Посчитать количество чисел с чётной длиной.

Пример: 3 123 4444 22

Результат: 2

16. Определить, сколько чисел из последовательности равны предыдущему числу.

Пример: 5 12 12 23 23 108

Результат: 2

17. Определить сумму всех цифр, которые делятся на 3, для всех чисел вместе.

Пример: 3 92 102 33

Результат: 15

18. Для заданного натурального числа N получите число, полученное реверсом его цифр, и вычислите модуль разности между исходным числом и его обратным представлением.

Пример: 2 123 400

Результат: 198 ($|123-321|=198$)

396 ($|400-4|=396$)

19. Определить, сколько чисел из последовательности равны предыдущему числу.

Пример: 5 12 12 23 23 108

Результат: 2

20. Вывести все числа, у которых сумма цифр состоит из одного десятичного разряда.

Пример: 4 19 93 44 29

Результат: 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: 3 27 14 99

Результат: 1 3

22. Необходимо посчитать количество нечётных цифр для каждого числа.

Пример: 3 12 5 444

Результат: 0 0 1

23. Посчитать количество чисел, являющихся полными квадратами (их корень является целым числом).

Пример: 4 55 81 32 49

Результат: 2

24. Вывести номера чисел из последовательности, у которых сумма цифр меньше их произведения (например, у числа 87 сумма цифр равна 15, а произведение 56).

Пример: 3 27 14 99

Результат: 1 3

25. Дана последовательность чисел. Определите количество чисел, у которых все цифры одинаковы.

Пример: 4 222 121 4444 123

Результат: 2

26. Все числа из последовательности, которые составлены из цифр, идущих по возрастанию необходимо перевернуть и вывести на экран.

Пример: 4 87 129 33 45

Результат: 921 54

27. Для каждого числа из последовательности определить, можно ли поставить внутри числа символ «/» таким образом, чтобы получилась корректная запись даты в формате день:месяц.

Пример: 3 1244 32 199

Результат: NOYESYES

28. Посчитать сумму всех чисел, у которых на нечётной позиции стоит нечётная цифра (нумерация позиций в числе идёт начиная с 0 справа налево).

Пример: 4 101 32 21 112

Результат: 144