

Министерство образования и науки РФ
Государственное образовательное учреждение
Высшего профессионального образования
Иркутский Государственный университет
Институт математики, экономики и информатики

Сборник задач по программированию с примерами.

2011

Сборник предназначен для студентов младших курсов специальностей и направлений «прикладная математика и информатика» и «математическое обеспечение и администрирование информационных систем».

Может использоваться в курсах «Практикум по информатике», «Алгоритмы и структуры компьютерной обработки данных».

Составитель: Мезенцев А.В.

© Иркутский Государственный
университет, 2011 г.

Оглавление.

1. Нахождение суммарных значений	5
2. Вычисление функции разложением ее в ряд.....	6
3. Целые числа	7
4. Одномерные массивы (1).....	8
5. Матрицы (1)	9
6. Строки	10
7. Одномерные массивы (2).....	11
8. Матрицы (2)	12
9. Визуальный вектор	13
10. Визуальная матрица.....	14
11. Многострочный текст.....	15
12. График функции	16
13. Редактирование ломанной.....	18
14. Файл последовательного доступа.....	19
15. Файл прямого доступа	20
16. Классы	20
17. Динамические структуры данных	23
Примеры на языке Java	26
18. Вычисление значения функции как суммы бесконечного ряда.....	26
19. Замена в строке одной подстроки другой	26
20. Поиск в строке всех вхождений заданной подстроки.....	27
21. В строке поменять местами две подстроки	28
22. Пример функции, возвращающей одномерный массив.....	30
23. Пример функции, возвращающей двумерный массив.....	31
24. Пример функции, возвращающей строку.....	32

25. Пример рекурсивной функции вычисляющей факториал.....	33
26. Пример рекурсивной функции вычисляющей максимум в одномерном массиве	33
27. Пример рекурсивной процедуры вычисляющей максимум в одномерном массиве	34
28. Поиск подпоследовательности в последовательности.....	35

1. Нахождение суммарных значений

1.1. Найти сумму расстояний от начала координат до точек гиперболы $y=a/x$, имеющих абсциссы 1, 2.1, 3.2, 4.3 и т.д. см, и среднее арифметическое этих n расстояний.

1.2. Найти суммарный объем n цилиндров высоты h , если радиусы их оснований образуют последовательность 3, 6, 9, ..., $3n$. Найти среднюю длину радиуса.

1.3. Найти суммарную площадь n колец, внутренние радиусы которых одинаковы и равны r , а значения внешних радиусов образуют последовательность 5, 10, 15, ..., $5n$. Найти среднюю длину внешнего радиуса.

1.4. Найти суммарную площадь n прямоугольных треугольников, если длина меньшего катета треугольника равна квадратному корню из длины большего, а длины больших катетов равны a , $a/2$, ..., a/n . Найти среднюю длину большого катета.

1.5. Найти сумму площадей n равносторонних треугольников, стороны которых равны a , $a\sqrt{2} + 1$, $a\sqrt{3} + 2$, ..., $a\sqrt{n} + n - 1$, и среднюю площадь.

1.6. В пирамиде, построенной из одинаковых кубиков, n слоев. В верхнем слое – 1 кубик, во втором – 4 кубика, в третьем – 9 и т.д. Найти общее число кубиков и их суммарный вес, если вес одного кубика равен a .

1.7. Найти суммарный путь, пройденный движущимся телом за n секунд, если за 1-ю секунду оно прошло путь, равный P , за 2-ю секунду – путь, равный $P\sqrt{2} + 1$, за 3-ю секунду - $P\sqrt{3} + 2$ и т.д. Найти путь, проходимый телом за одну секунду в среднем.

1.8. Найти общий объем n прямоугольных параллелепипедов, у каждого из которых одинаковое соотношение длин ребер, исходящих из одного угла: 1:2:3, если длины их наименьших ребер образуют последовательность a , $a/2$, $a/3$, ..., a/n . Найти среднюю длину наименьшего ребра.

1.9. Найти сумму расстояний от начала координат до точек параболы $y=ax^2 + b$, имеющих абсциссы 1, 2, 3, ..., n см, и среднее арифметическое этих n расстояний.

1.10. Игрушка состоит из n матрешек, вложенных друг в друга, причем объем наименьшей из них равен V . Объемы других матрешек образуют последовательность $(1.3)^3 V$, $(1.6)^3 V$, $(1.9)^3 V$, Найти суммарный объем матрешек и средний объем.

1.11. Найти суммарный объем n шаров, если их радиусы образуют последовательность r , $r + 0.2$, $r + 0.4$, $r + 0.6$, Найти средний объем шара.

1.12. Найти суммарный объем n конусов с высотой h и радиусами оснований $0.3^2, 0.6^2, 0.9^2$ и т.д. Найти средний объем конуса.

1.13. Найти суммарную длину n отрезков, длины которых образуют последовательность $a, a(\sqrt{2} + 1), \dots, a(\sqrt{n} + n - 1)$. Найти среднюю длину отрезка.

2. Вычисление функции разложением ее в ряд

Написать программу нахождения суммы ряда с заданной точностью ϵ . Использовать рекуррентные соотношения при вычислении очередного члена ряда. Вычисление заканчивается, если модуль очередного слагаемого становится меньше заданного значения точности (ϵ), причем для этих рядов (при $|x| < 1$) абсолютная величина суммы всех отброшенных членов ряда будет меньше ϵ . Для оценки правильности расчетов предусмотреть вычисление по контрольной формуле.

№	Ряд	Контрольная формула
2.1	$x - \frac{2}{6}x^2 + \frac{2 \cdot 5}{6 \cdot 9}x^3 - \dots \pm \frac{2 \cdot 5 \cdot \dots \cdot (3i-4)}{6 \cdot 9 \cdot \dots \cdot 3i}x^i \mp \dots$	$3\sqrt[3]{1+x} - 3$
2.2	$\frac{x(2+x)}{2!} - \frac{x^3(4+x)}{4!} + \frac{x^5(6+x)}{6!} - \dots$ $\dots \pm \frac{x^{2i-1}(2i+x)}{(2i)!} \mp \dots$	$\sin x - \cos x + 1$
2.3	$\frac{1}{4}x - \frac{1 \cdot 5}{4 \cdot 8}x^2 + \frac{1 \cdot 5 \cdot 9}{4 \cdot 8 \cdot 12}x^3 - \dots \pm \frac{1 \cdot 5 \cdot 9 \cdot \dots \cdot (4i-3)}{4 \cdot 8 \cdot 12 \cdot \dots \cdot 4i}x^i \mp \dots$	$1 - \frac{1}{\sqrt[4]{1+x}}$
2.4	$\frac{3x^2}{4!} - \frac{5x^4}{6!} + \frac{7x^6}{8!} - \frac{9x^8}{10!} + \dots \pm \frac{(2i+1)x^{2i}}{(2i+2)!} \mp \dots$	$\frac{1 - \cos x - x \sin x}{x^2} + 0.5$
2.5	$\frac{1}{3}x - \frac{1 \cdot 4}{3 \cdot 6}x^2 + \frac{1 \cdot 4 \cdot 7}{3 \cdot 6 \cdot 9}x^3 - \dots \pm \frac{1 \cdot 4 \cdot 7 \cdot \dots \cdot (3i-2)}{3 \cdot 6 \cdot 9 \cdot \dots \cdot 3i}x^i \mp \dots$	$1 - \frac{1}{\sqrt[3]{1+x}}$
2.6	$\frac{1}{x^2} + \frac{x^2}{4!} - \frac{x^4}{6!} + \frac{x^6}{8!} - \frac{x^8}{10!} + \dots \pm \frac{x^{2i}}{(2i+2)!} \mp \dots$	$\frac{\cos(x)}{x^2} + \frac{1}{2}$
2.7	$1 - \frac{3}{2}x + \frac{3 \cdot 5}{2 \cdot 4}x^2 - \frac{3 \cdot 5 \cdot 7}{2 \cdot 4 \cdot 6}x^3 + \dots \pm \frac{3 \cdot 5 \cdot 7 \cdot \dots \cdot (2i+1)}{2 \cdot 4 \cdot 6 \cdot \dots \cdot 2i}x^i \mp \dots$	$\frac{1}{\sqrt{(1+x)^3}}$

2.8	$\frac{(2x)^2}{2!} - \frac{(2x)^4}{4!} + \frac{(2x)^6}{6!} - \dots \pm \frac{(2x)^{2i}}{(2i)!} \mp \dots$	$2\sin^2 x$
2.9	$\frac{1}{2}x - \frac{1 \cdot 3}{2 \cdot 4}x^2 + \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6}x^3 - \dots \pm \frac{1 \cdot 3 \cdot 5 \cdot \dots \cdot (2i-1)}{2 \cdot 4 \cdot 6 \cdot \dots \cdot 2i}x^i \mp \dots$	$1 - \frac{1}{\sqrt{1+x}}$
2.10	$\frac{x}{3!} - \frac{x^3}{5!} + \frac{x^5}{7!} - \frac{x^7}{9!} + \dots \pm \frac{x^{2i-1}}{(2i+1)!} \mp \dots$	$\frac{x - \sin x}{x^2}$
2.11	$1 - \frac{5}{2}x + \frac{5 \cdot 7}{2 \cdot 4}x^2 - \frac{5 \cdot 7 \cdot 9}{2 \cdot 4 \cdot 6}x^3 + \dots \pm \frac{5 \cdot 7 \cdot 9 \cdot \dots \cdot (2i+3)}{2 \cdot 4 \cdot 6 \cdot \dots \cdot 2i}x^i \mp \dots$	$\frac{1}{\sqrt{(1+x)^5}}$
2.12	$\frac{2x}{1!} - \frac{3x^2}{2!} + \frac{4x^3}{3!} - \frac{5x^4}{4!} + \dots \pm \frac{(i+1)x^i}{i!} \mp \dots$	$x \cdot e^{-x} - e^{-x} + 1$
2.13	$x - \frac{1}{4}x^2 + \frac{1 \cdot 3}{4 \cdot 6}x^3 - \frac{1 \cdot 3 \cdot 5}{4 \cdot 6 \cdot 8}x^4 + \dots \pm \frac{1 \cdot 3 \cdot 5 \cdot \dots \cdot (2i-3)}{4 \cdot 6 \cdot 8 \cdot \dots \cdot 2i}x^i \mp \dots$	$2\sqrt{1+x} - 2$

3. Целые числа

- 3.1. Проверить, является ли данное натуральное число числом Армстронга. Натуральное число из n цифр называется числом Армстронга, если сумма его цифр, возведенных в n -ю степень, равна самому числу, например, $153=1^3+5^3+3^3$.
- 3.2. Проверить, является ли данное натуральное число числом Фибоначчи.
- 3.3. Проверить, является ли данное натуральное число палиндромом (перевертышем). Натуральное число называется палиндромом, если его запись читается одинаково с начала и с конца, например, 2332, 313, 6.
- 3.4. Две точки на плоскости заданы своими координатами. Выяснить, образуют ли эти точки вместе с центром координат прямоугольный треугольник.
- 3.5. Даны два натуральных числа. Выяснить, имеют ли они хотя бы два общих множителя.
- 3.6. Проверить, является ли данное натуральное число числом Мерсена. Простое число называется числом Мерсена, если оно может быть представлено в виде $2^p - 1$, где p – тоже простое число.
- 3.7. Проверить, можно ли представить данное натуральное число в виде суммы квадратов двух натуральных чисел.
- 3.8. Проверить, являются ли все цифры данного натурального числа различными.

- 3.9. Дано натуральное число. Можно ли его представить в виде произведения трех последовательных целых чисел.
- 3.10. Даны два натуральных числа. Выяснить, являются ли они взаимно простыми. Натуральные числа называются взаимно простыми, если наибольший общий делитель этих чисел равен 1.
- 3.11. Даны два натуральных числа. Выяснить, являются ли они дружественными. Два натуральных числа называют дружественными, если каждое из них равно сумме всех делителей другого, кроме самого этого числа.
- 3.12. Дробь задана двумя натуральными числами – числителем и знаменателем. Выяснить, является ли эта дробь несократимой.
- 3.13. Проверить, является ли данное натуральное число простым.

4. Одномерные массивы (1).

- 4.1. Дана последовательность целых чисел a_1, a_2, \dots, a_n . Выяснить, является ли она возрастающей последовательностью простых чисел.
- 4.2. Даны действительные числа c_1, c_2, \dots, c_n . Найти произведение среднего арифметического положительных чисел и среднего арифметического отрицательных чисел.
- 4.3. Даны действительные числа c_1, c_2, \dots, c_n . Найти произведение суммы чисел с четными индексами и суммы чисел с нечетными индексами.
- 4.4. Даны целые числа a_1, a_2, \dots, a_n . Выяснить, есть ли среди них повторяющиеся числа, и найти среднее арифметическое без учета повторов. Например, для последовательности чисел 1, 3, 4, 3, 6, 3, 0 надо найти среднее арифметическое чисел 1, 3, 4, 6, 0.
- 4.5. Даны целые числа a_1, a_2, \dots, a_n . Пусть M - наибольшее, а m - наименьшее этих чисел. Получить в порядке возрастания все целые из интервала (m, M) , которые не входят в последовательность a_1, a_2, \dots, a_n .
- 4.6. Даны две последовательности целых чисел a_1, a_2, \dots, a_n и c_1, c_2, \dots, c_k , $n \geq k$. Выяснить, является ли вторая последовательность подпоследовательностью первой. Например, последовательность 4, 6, 3 является подпоследовательностью последовательности 0, 2, 4, 6, 3, -1, 1.
- 4.7. Дана последовательность чисел a_1, a_2, \dots, a_n . Найти положительную подпоследовательность наибольшей длины.
- 4.8. Дана последовательность чисел a_1, a_2, \dots, a_n . Выяснить, сколько чисел и какие входят в последовательность более чем по одному разу.
- 4.9. Дана последовательность чисел a_1, a_2, \dots, a_n . Определить число элементов этой последовательности, не превышающих среднего арифметического положительных элементов.

- 4.10. В последовательности чисел a_1, a_2, \dots, a_n найти два числа, среднее арифметическое которых ближе всего к числу y .
- 4.11. Дана последовательность целых чисел a_1, a_2, \dots, a_n . Выяснить, является ли она симметричной последовательностью простых чисел.
- 4.12. Дана последовательность целых чисел a_1, a_2, \dots, a_n . Если все числа являются четными, поменять местами элементы a_1 и a_n, a_2 и a_{n-1} и т.д., в противном случае нечетные элементы последовательности обнулить.
- 4.13. Даны действительные числа $a, b, c_1, c_2, \dots, c_n$. Вычислить сумму и среднее арифметическое тех элементов c_1, c_2, \dots, c_n , которые принадлежат отрезку $[a, b]$.

5. Матрицы (1).

- 5.1. Даны два n – мерных массива x и y , которые задают координаты n точек на плоскости. Найти наиболее близкие друг другу точки.
- 5.2. Дана матрица A размера $m \times n$ и два действительных числа a и b . Построить вектор c с компонентами c_1, c_2, \dots, c_n , где c_i – количество элементов i -го столбца матрицы, не принадлежащих отрезку $[a, b]$.
- 5.3. В квадратной матрице найти среднее арифметическое положительных элементов главной диагонали и среднее арифметическое отрицательных элементов побочной диагонали.
- 5.4. Дана матрица A размера $m \times n$. Найти количество строк матрицы, элементы которых образуют симметричные последовательности (палиндромы).
- 5.5. Дана матрица A размера $m \times n$. Найти количество строк матрицы, элементы которых образуют монотонно возрастающую последовательность.
- 5.6. Для квадратной матрицы найти среднее арифметическое элементов над главной диагональю и количество четных элементов под ней.
- 5.7. Для квадратной матрицы найти разность сумм элементов над и под главной диагональю.
- 5.8. Дана матрица A размера $m \times n$. Найти количество строк матрицы, сумма элементов которых отрицательна.
- 5.9. Даны две квадратные матрицы : A размера $m \times n$ и B размера $p \times q$, $m > p$, $n > q$. Проверить, является ли матрица A клеткой матрицы B .
- 5.10. Дана целочисленная квадратная матрица. Написать программу упорядочения ее строк по возрастанию сумм их элементов.
- 5.11. Дана целочисленная квадратная матрица. Проверить, является ли она магическим квадратом. (Магический квадрат – квадратная таблица размера $n \times n$, составленная из чисел $1, 2, 3, \dots, n^2$ так, что суммы по каждому столбцу, каждой строке и каждой из двух диагоналей равны между собой.)

5.12. Дана целочисленная квадратная матрица. Проверить, является ли она латинским квадратом. (Латинский квадрат – квадратная таблица размера $n \times n$, каждая строка и каждый столбец которой содержит числа $1, 2, \dots, n$.)

5.13. В прямоугольной матрице найти наибольший из минимальных элементов строк матрицы.

6. Строки.

6.1. Дана строка символов. Слова в строке отделяются друг от друга одним пробелом. Вывести самое длинное слово.

6.2. Дана строка символов. Слова в строке отделяются друг от друга одним пробелом. Вывести самое короткое слово.

6.3. Дана строка символов. Слова в строке отделяются друг от друга одним пробелом. Написать программу, вычисляющую среднюю длину слов в строке.

6.4. Дана строка символов. Слова в строке отделяются друг от друга одним пробелом. Удалить из строки самое длинное слово.

6.5. Дана строка символов. Слова в строке отделяются друг от друга одним пробелом. Удалить из строки самое короткое слово.

6.6. Дана строка символов. Слова в строке отделяются друг от друга одним пробелом. Заменить в строке одно заданное слово (если оно есть) другим.

6.7. Дана строка символов. Слова в строке отделяются друг от друга любым количеством пробелов. Преобразовать строку таким образом, чтобы слова отделялись строго одним пробелом.

6.8. Дана строка символов. Слова в строке отделяются одним пробелом. Поменять местами самое длинное и самое короткое слово.

6.9. Дана строка из нескольких слов. Слова отделяются друг от друга пробелами или запятыми. Подсчитать количество слов, длина которых больше заданного числа.

6.10. Дана строка из нескольких слов. Слова отделяются друг от друга пробелами или запятыми. Подсчитать количество слов, длина которых меньше заданного числа.

6.11. Дана строка из нескольких слов. Слова отделяются друг от друга пробелами или запятыми. Вывести все слова, длина которых больше заданного числа.

6.12. Дана строка из нескольких слов. Слова отделяются друг от друга пробелами или запятыми. Вывести все слова, длина которых меньше заданного числа.

6.13. Дана строка из нескольких слов. Слова отделяются друг от друга пробелами или запятыми. Подсчитать количество слов начинающихся и заканчивающихся одной и той же буквой.

6.14. Дана строка из нескольких слов. Слова отделяются друг от друга пробелами или запятыми. Вывести слова, начинающиеся и заканчивающиеся одной и той же буквой.

7. Одномерные массивы (2).

7.1. Дан массив целых чисел $\{a_i\} (i=1, 2, \dots, n)$. Написать программу, проверки упорядочены ли значения компонентов массива по неубыванию.

7.2. Дан массив целых чисел $\{a_i\} (i=1, 2, \dots, n)$. Написать программу, проверки упорядочены ли значения компонентов массива по убыванию.

7.3. Даны два массива вещественных чисел $\{a_i\} (i=1, 2, \dots, n), \{b_i\} (i=1, 2, \dots, n)$. Интерпретируя компоненты этих массивов как границы отрезков $[a_i, b_i]$ на числовой прямой, вычислить длину той части числовой прямой, которая покрывается этими отрезками.

7.4. Даны два массива вещественных чисел $\{a_i\} (i=1, 2, \dots, n), \{b_i\} (i=1, 2, \dots, n)$ – координаты двух векторов n -мерного пространства исходящих из начала координат. Выяснить, как соотносятся длины этих двух векторов.

7.5. Дан массив целых чисел $\{a_i\} (i=1, 2, \dots, n)$ и целое число x . Значения компонентов массива упорядочены по неубыванию. Написать программу вычисления такого номера компонента массива i (если он существует), что $a_i \leq x \leq a_{i+1}$.

7.6. Дан массив целых чисел $\{a_i\} (i=1, 2, \dots, n)$, целые c и d ($c < d$). Написать программу вычисления количества компонентов массива, значения которых принадлежат отрезку $[c, d]$.

7.7. Дан массив целых чисел $\{a_i\} (i=1, 2, \dots, n)$, целое число k ($k > 0$). Осуществить циклический сдвиг значений компонентов массива вправо на k компонент.

7.8. Дан массив целых чисел $\{a_i\} (i=1, 2, \dots, n)$, целое число k ($k > 0$). Осуществить циклический сдвиг значений компонентов массива влево на k компонент.

7.9. Дан массив целых чисел $\{a_i\} (i=1, 2, \dots, n)$. Поменять местами значения компонентов массива таким образом, чтобы сначала следовали все отрицательные значения, а затем все неотрицательные с сохранением порядка их следования в исходном массиве.

7.10. Пусть значения границ n отрезков $[a_i, b_i]$ ($1 \leq i \leq n$) числовой прямой заданы в виде двух массивов $\{a_i\}$ и $\{b_i\}$. Упорядочить отрезки по убыванию их длин.

7.11. Пусть коэффициенты полинома $P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ заданы в виде массива вещественных чисел $\{a_i\}$ ($0 \leq i \leq n$). Вычислить значение полинома в точке x .

7.12. Дан массив целых чисел $\{a_i\}$ ($i=1, 2, \dots, n$). Написать программу определения что больше, наибольший среди компонентов с положительными значениями, или модуль наименьшего среди компонентов с отрицательными значениями.

7.13. Дан массив целых чисел $\{a_i\}$ ($i=1, 2, \dots, n$). Написать программу определения что больше, наименьший среди компонентов с положительными значениями, или модуль наибольшего среди компонентов с отрицательными значениями.

7.14. Дан массив целых чисел $\{a_i\}$ ($i=1, 2, \dots, n$), целые c и d ($c < d$). Написать программу определения что больше, количество компонентов массива принадлежащих отрезку $[c, d]$, или количество компонентов массива не принадлежащих этому отрезку.

8. Матрицы (2).

8.1. Дана целочисленная квадратная матрица $\{a_{ij}\}$ ($1 \leq i \leq n, 1 \leq j \leq n$). Написать программу, вычисляющую наименьшее значение среди компонентов матрицы расположенных ниже главной диагонали.

8.2. Дана целочисленная квадратная матрица $\{a_{ij}\}$ ($1 \leq i \leq n, 1 \leq j \leq n$). Написать программу, вычисляющую наименьшее значение среди компонентов матрицы расположенных выше главной диагонали.

8.3. Дана целочисленная квадратная матрица $\{a_{ij}\}$ ($1 \leq i \leq n, 1 \leq j \leq n$). Написать программу, вычисляющую наибольшее значение среди компонентов матрицы расположенных ниже главной диагонали.

8.4. Дана целочисленная квадратная матрица $\{a_{ij}\}$ ($1 \leq i \leq n, 1 \leq j \leq n$). Написать программу, вычисляющую наибольшее значение среди компонентов матрицы расположенных выше главной диагонали.

8.5. Дана целочисленная квадратная матрица $\{a_{ij}\}$ ($1 \leq i \leq n, 1 \leq j \leq n$). Написать программу, вычисляющую наименьшее значение среди компонентов матрицы расположенных выше главной и выше побочной диагоналей.

8.6. Дана целочисленная квадратная матрица $\{a_{ij}\}$ ($1 \leq i \leq n, 1 \leq j \leq n$). Написать программу, вычисляющую наименьшее значение среди компонентов матрицы расположенных выше главной и ниже побочной диагоналей.

- 8.7. Дана целочисленная квадратная матрица $\{a_{ij}\}$ ($1 \leq i \leq n, 1 \leq j \leq n$). Написать программу вычисляющую сумму компонентов матрицы расположенных выше главной диагонали.
- 8.8. Дана целочисленная квадратная матрица $\{a_{ij}\}$ ($1 \leq i \leq n, 1 \leq j \leq n$). Написать программу вычисляющую сумму компонентов матрицы расположенных ниже главной диагонали.
- 8.9. Дана целочисленная квадратная матрица $\{a_{ij}\}$ ($1 \leq i \leq n, 1 \leq j \leq n$). Написать программу вычисляющую сумму компонентов матрицы расположенных выше побочной диагонали.
- 8.10. Дана целочисленная квадратная матрица $\{a_{ij}\}$ ($1 \leq i \leq n, 1 \leq j \leq n$). Написать программу вычисляющую сумму компонентов матрицы расположенных ниже побочной диагонали.
- 8.11. Дана целочисленная квадратная матрица $\{a_{ij}\}$ ($1 \leq i \leq n, 1 \leq j \leq n$). Написать программу вычисляющую сумму компонентов матрицы расположенных выше главной и выше побочной диагоналей.
- 8.12. Дана целочисленная квадратная матрица $\{a_{ij}\}$ ($1 \leq i \leq n, 1 \leq j \leq n$). Написать программу вычисляющую сумму компонентов матрицы расположенных выше главной и ниже побочной диагоналей.
- 8.13. Дана целочисленная квадратная матрица $\{a_{ij}\}$ ($1 \leq i \leq n, 1 \leq j \leq n$). Написать программу вычисляющую сумму компонентов матрицы расположенных ниже главной и выше побочной диагоналей.

9. Визуальный вектор.

- 9.1. Даны действительные числа a_1, \dots, a_n . Найти подпоследовательность наибольшей длины расположенную между соседними локальными экстремумами.
- 9.2. Даны действительные числа a_1, \dots, a_n . Найти подпоследовательность наименьшей длины расположенную между соседними локальными экстремумами.
- 9.3. Даны действительные числа a_1, \dots, a_n . Найти подпоследовательность наибольшей длины расположенную между соседними локальными максимумами.
- 9.4. Даны действительные числа a_1, \dots, a_n . Найти подпоследовательность наименьшей длины расположенную между соседними локальными максимумами.
- 9.5. Даны действительные числа a_1, \dots, a_n . Найти подпоследовательность наибольшей длины расположенную между соседними локальными минимумами.

- 9.6. Даны действительные числа a_1, \dots, a_n . Найти подпоследовательность наименьшей длины расположенную между соседними локальными минимумами.
- 9.7. Даны действительные числа a_1, \dots, a_n . Найти наибольшую положительную подпоследовательность (т. е. состоящую только из положительных чисел).
- 9.8. Даны действительные числа a_1, \dots, a_n . Найти наименьшую положительную подпоследовательность (т. е. состоящую только из положительных чисел).
- 9.9. Даны действительные числа a_1, \dots, a_n . Найти наибольшую отрицательную подпоследовательность (т. е. состоящую только из отрицательных чисел).
- 9.10. Даны действительные числа a_1, \dots, a_n . Найти наименьшую отрицательную подпоследовательность (т. е. состоящую только из отрицательных чисел).
- 9.11. Даны действительные числа a_1, \dots, a_n . Найти подпоследовательность наибольшей длины расположенную между членами с нулевыми значениями.
- 9.12. Даны действительные числа a_1, \dots, a_n . Найти подпоследовательность наименьшей длины расположенную между членами с нулевыми значениями.

10. Визуальная матрица.

- 10.1. Дана вещественная матрица $\{a_{ij}\}$ ($1 \leq i \leq n$, $1 \leq j \leq m$). Написать программу, выделяющую локальные максимумы, глобальный максимум в каждой строке и, отдельно, глобальный максимум в матрице.
- 10.2. Дана вещественная матрица $\{a_{ij}\}$ ($1 \leq i \leq n$, $1 \leq j \leq m$). Написать программу, выделяющую локальные минимумы, глобальный минимум в каждой строке и, отдельно, глобальный минимум в матрице.
- 10.3. Дана вещественная матрица $\{a_{ij}\}$ ($1 \leq i \leq n$, $1 \leq j \leq m$). Написать программу, выделяющую локальные максимумы, глобальный максимум в каждом столбце и, отдельно, глобальный максимум в матрице.
- 10.4. Дана вещественная матрица $\{a_{ij}\}$ ($1 \leq i \leq n$, $1 \leq j \leq m$). Написать программу, выделяющую локальные минимумы, глобальный минимум в каждом столбце и, отдельно, глобальный минимум в матрице.
- 10.5. Дана вещественная матрица $\{a_{ij}\}$ ($1 \leq i \leq n$, $1 \leq j \leq n$). Написать программу, выделяющую локальные максимумы, глобальный максимум в каждой диагонали параллельной главной и, отдельно, глобальный максимум в матрице.
- 10.6. Дана вещественная матрица $\{a_{ij}\}$ ($1 \leq i \leq n$, $1 \leq j \leq n$). Написать программу, выделяющую локальные минимумы, глобальный минимум в каждой диагонали параллельной главной и, отдельно, глобальный минимум в матрице.
- 10.7. Дана вещественная матрица $\{a_{ij}\}$ ($1 \leq i \leq n$, $1 \leq j \leq n$). Написать программу, выделяющую локальные максимумы, глобальный максимум в каждой

диагонали параллельной побочной и, отдельно, глобальный максимум в матрице.

10.8. Дана вещественная матрица $\{a_{ij}\}$ ($1 \leq i \leq n$, $1 \leq j \leq n$). Написать программу, выделяющую локальные минимумы, глобальный минимум в каждой диагонали параллельной побочной и, отдельно, глобальный минимум в матрице.

10.9. Дана вещественная матрица $\{a_{ij}\}$ ($1 \leq i \leq n$, $1 \leq j \leq n$). Своими главной и побочной диагоналями матрица разбита на четыре треугольные части. Написать программу, выделяющую в каждом из треугольников максимум и, отдельно, наименьший из них.

10.10. Дана вещественная матрица $\{a_{ij}\}$ ($1 \leq i \leq n$, $1 \leq j \leq n$). Своими главной и побочной диагоналями матрица разбита на четыре треугольные части. Написать программу, выделяющую в каждом из треугольников минимум и, отдельно, наибольший из них.

10.11. Дана вещественная матрица $\{a_{ij}\}$ ($1 \leq i \leq n$, $1 \leq j \leq n$). Своими главной и побочной диагоналями матрица разбита на четыре треугольные части. Написать программу, выделяющую в каждом из треугольников максимум и, отдельно, максимум в матрице.

10.12. Дана вещественная матрица $\{a_{ij}\}$ ($1 \leq i \leq n$, $1 \leq j \leq n$). Своими главной и побочной диагоналями матрица разбита на четыре треугольные части. Написать программу, выделяющую в каждом из треугольников минимум и, отдельно, минимум в матрице.

10.13. Дана вещественная матрица $\{a_{ij}\}$ ($1 \leq i \leq n$, $1 \leq j \leq n$). Среди диагоналей параллельных главной диагонали выделить ту, сумма элементов которой наибольшая.

10.14. Дана вещественная матрица $\{a_{ij}\}$ ($1 \leq i \leq n$, $1 \leq j \leq n$). Среди диагоналей параллельных главной диагонали выделить ту, сумма элементов которой наименьшая.

10.15. Дана вещественная матрица $\{a_{ij}\}$ ($1 \leq i \leq n$, $1 \leq j \leq n$). Среди диагоналей параллельных побочной диагонали выделить ту, сумма элементов которой наибольшая.

10.16. Дана вещественная матрица $\{a_{ij}\}$ ($1 \leq i \leq n$, $1 \leq j \leq n$). Среди диагоналей параллельных побочной диагонали выделить ту, сумма элементов которой наименьшая.

11. Многострочный текст.

В текстовом окне задан многострочный текст. Слова отделяются друг от друга пробелами. Каждое предложение в тексте заканчивается точкой.

- 11.1. Последовательно выделять слова, где гласных букв больше чем согласных.
- 11.2. Последовательно выделять слова, где согласных букв больше чем гласных.
- 11.3. Последовательно выделять предложения с четным количеством слов.
- 11.4. Последовательно выделять предложения с нечетным количеством слов.
- 11.5. Последовательно выделять слова с удвоенной гласной.
- 11.6. Последовательно выделять строки с четным количеством слов.
- 11.7. Последовательно выделять строки с нечетным количеством слов.
- 11.8. Последовательно выделять самое короткое слово в каждом предложении.
- 11.9. Последовательно выделять самое длинное слово в каждом предложении.
- 11.10. Последовательно выделять слова содержащие удвоенные согласные буквы.
- 11.11. Последовательно выделять слова длина которых больше заданного числа.
- 11.12. Последовательно выделять слова содержащие заданную подстроку.
- 11.13. Последовательно выделять слова длина которых меньше заданного числа.

12. График функции.

Написать программу построения графика заданной функции на заданном отрезке $[a;b]$. График должен содержать оси, оцифровку значений по осям. Единицы масштаба по осям X и Y должны совпадать (для контроля необходимо вывести график функции $y = x$. Все поле рисования должно быть разграфлено как тетрадь в клетку (каким-либо нейтральным цветом, например светло зеленым). Программа должна допускать построение графика функции с другими заданными коэффициентами (например, если основная функция $\sin x$, то программа должна допускать построение функции $a \cdot \sin(bx+c)+d$. По умолчанию $a=1, b=1, c=0, d=0$).

- 12.1. Написать программу построения графика функции $x + \sin x$ на заданном отрезке $[a;b]$.
- 12.2. Написать программу построения графика функции $x + \cos x$ на заданном отрезке $[a;b]$.
- 12.3. Написать программу построения графика функции $1/\sin x$ на заданном отрезке $[a;b]$.
- 12.4. Написать программу построения графика функции $1/\cos x$ на заданном отрезке $[a;b]$.

- 12.5. Написать программу построения графика функции $x/\sin x$ на заданном отрезке $[a;b]$.
- 12.6. Написать программу построения графика функции $x/\cos x$ на заданном отрезке $[a;b]$.
- 12.7. Написать программу построения графика функции $e^{\sin x}$ на заданном отрезке $[a;b]$.
- 12.8. Написать программу построения графика функции $e^{\cos x}$ на заданном отрезке $[a;b]$.
- 12.9. Написать программу построения графика функции $e^{\sin x + \cos x}$ на заданном отрезке $[a;b]$.
- 12.10. Написать программу построения графика функции $\ln |\sin x|$ на заданном отрезке $[a;b]$.
- 12.11. Написать программу построения графика функции $\ln |\cos x|$ на заданном отрезке $[a;b]$.
- 12.12. Написать программу построения графика функции $\ln |\sin x + \cos x|$ на заданном отрезке $[a;b]$.
- 12.13. Написать программу построения графика функции $1/\ln |\sin x|$ на заданном отрезке $[a;b]$.
- 12.14. Написать программу построения графика функции $1/e^{\cos x}$ на заданном отрезке $[a;b]$.
- 12.15. Написать программу построения графика функции $1/e^{\sin x}$ на заданном отрезке $[a;b]$.
- 12.16. Написать программу построения графика функции $x/e^{\cos x}$ на заданном отрезке $[a;b]$.
- 12.17. Написать программу построения графика функции $x/e^{\sin x}$ на заданном отрезке $[a;b]$.
- 12.18. Написать программу построения графика функции $x/e^{\sin x + \cos x}$ на заданном отрезке $[a;b]$.
- 12.19. Написать программу построения графика функции $1/e^{\sin x + \cos x}$ на заданном отрезке $[a;b]$.
- 12.20. Написать программу построения графика функции $\ln |x + \sin x|$ на заданном отрезке $[a;b]$.
- 12.21. Написать программу построения графика функции $\ln |x + \cos x|$ на заданном отрезке $[a;b]$.
- 12.22. Написать программу построения графика функции $x + \operatorname{tg} x$ на заданном отрезке $[a;b]$.

12.23. Написать программу построения графика функции $x + ctg x$ на заданном отрезке $[a; b]$.

12.24. Написать программу построения графика функции $x + e^{\sin x}$ на заданном отрезке $[a; b]$.

12.25. Написать программу построения графика функции $x + e^{\cos x}$ на заданном отрезке $[a; b]$.

12.26. Написать программу построения графика функции $x + e^{\sin x + \cos x}$ на заданном отрезке $[a; b]$.

12.27. Написать программу построения графика функции $x / e^{|\sin x|}$ на заданном отрезке $[a; b]$.

12.28. Написать программу построения графика функции $x / e^{|\cos x|}$ на заданном отрезке $[a; b]$.

12.29. Написать программу построения графика функции $x / e^{|\sin x + \cos x|}$ на заданном отрезке $[a; b]$.

12.30. Написать программу построения графика функции $x - x / e^{|\sin x|}$ на заданном отрезке $[a; b]$.

12.31. Написать программу построения графика функции $|x / e^{\sin x}|^{1/2}$ на заданном отрезке $[a; b]$.

12.32. Написать программу построения графика функции $|x / e^{|\sin x|}|^{1/2}$ на заданном отрезке $[a; b]$.

13. Редактирование ломанной.

Написать программу графического ввода и редактирования ломанной (многоугольника). Программа должна иметь следующие функции:

- пользователь нажал левую кнопку мыши – нарисовалась точка; еще одно нажатие приводит к рисованию еще одной точки и линии соединяющей две точки и т.д. (в задачах с многоугольником первая и последняя точки должны совпадать).
- пользователь нажал правую кнопку мыши (с каким-либо модификатором на клавиатуре) должна выполняться одна из операций (в зависимости от модификатора):
 - - удаление точки на которой стоит курсор;
 - - добавление точки в конец ломанной;
 - - добавление точки на отрезок, к которому эта точка ближе всего;
 - - перемещение существующей точки.

После редактирования ломанной (многоугольника) выделение должно сниматься.

13.1. Выделить самый короткий отрезок ломанной.

- 13.2. Выделить самый длинный отрезок ломанной.
- 13.3. Выделить самое короткое ребро многоугольника.
- 13.4. Выделить самое длинное ребро многоугольника.
- 13.5. Выделить самый большой внутренний угол выпуклого многоугольника.
- 13.6. Выделить самый маленький внутренний угол выпуклого многоугольника.
- 13.7. Выделить самый близкий к заданной точке отрезок ломанной.
- 13.8. Выделить самый дальний к заданной точке отрезок ломанной.
- 13.9. Выделить самое близкое к заданной точке ребро многоугольника.
- 13.10. Выделить самое дальнее к заданной точке ребро многоугольника.
- 13.11. Выделить самую короткую диагональ выпуклого многоугольника.
- 13.12. Выделить самую длинную диагональ выпуклого многоугольника.
- 13.13. В выпуклом многоугольнике его диагонали делят многоугольник на треугольники. Не рисуя лишних диагоналей выделить треугольник с наименьшей площадью.
- 13.14. В выпуклом многоугольнике его диагонали делят многоугольник на треугольники. Не рисуя лишних диагоналей выделить треугольник с наибольшей площадью.

14. Файл последовательного доступа.

Программа должна допускать следующие режимы работы:

- (1) Ввести числа и записать их в файл
- (2) Прочитать все числа из файла и вывести их
- (3) Выполнить над числами из файла заданные действия и вывести результат

- 14.1. Вычислить сумму всех чисел.
- 14.2. Найти наибольшее число.
- 14.3. Найти наименьшее число.
- 14.4. Вычислить сумму положительных чисел.
- 14.5. Вычислить сумму отрицательных чисел.
- 14.6. Найти наибольшее число среди положительных.
- 14.7. Найти наибольшее число среди отрицательных.
- 14.8. Найти наименьшее число среди положительных.
- 14.9. Найти наименьшее число среди отрицательных.
- 14.10. Вычислить количество чисел принадлежащих заданному отрезку.

- 14.11. Вычислить количество чисел не принадлежащих заданному отрезку.
- 14.12. Вычислить границы отрезка минимальной длины, которому принадлежат все числа.
- 14.13. Вычислить среднее арифметическое положительных чисел.

15. Файл прямого доступа.

Информация о группе студентов хранится в файле прямого доступа. Структура записи имеет следующий вид:

1. Фамилия
2. Имя
3. Отчество
4. Год рождения
5. Пол
6. Массив оценок (не менее трех предметов)

В группе не менее 10 человек (5 юношей и 5 девушек).

Программа должна допускать следующие режимы работы:

- (1) Добавить запись в файл (если файла нет, то создать его)
- (2) Просмотреть все записи файла
- (3) Вывести результаты запроса

- 15.1. Вывести информацию о самом старшем студенте
- 15.2. Вывести информацию о самом молодом студенте
- 15.3. Вывести информацию о самой старшей студентке
- 15.4. Вывести информацию о самой молодой студентке
- 15.5. Вывести информацию о самом успевающем студенте
- 15.6. Вывести информацию о самом неуспевающем студенте
- 15.7. Вывести информацию о самой успевающей студентке
- 15.8. Вывести информацию о самой неуспевающей студентке
- 15.9. Вывести список студенток получающих стипендию
- 15.10. Вывести список студенток не получающих стипендию
- 15.11. Вывести список студентов получающих стипендию
- 15.12. Вывести список студентов не получающих стипендию
- 15.13. Вывести список студентов отличников

16. Классы.

16.1. Вектор.

Хранит координаты конца вектора в n -мерном пространстве выходящего из начала координат. Методы: увеличение количества координат, уменьшение количества координат, сумма двух векторов, произведение двух векторов, разность двух векторов, умножение вектора на скаляр, длина вектора.

16.2. Строка.

Хранит строку переменной длины. Методы: конкатенация двух строк, вырезание подстроки, сравнение двух строк на равенство, неравенство, меньше, больше.

16.3. Дата.

Хранит день, месяц и год. Методы: разность в днях двух дат, дата отстоящая от заданной на заданное количество дней (месяцев), день недели.

16.4. Время.

Хранит секунды минуты и часы. Методы: разность в секундах двух отрезков времени, время отстоящее от заданного на заданное количество секунд (минут).

16.5. Множество.

Хранит элементы множества целого типа. Методы: принадлежность элемента множеству, добавить элемент, удалить элемент, является ли множество подмножеством данного, равны (не равны) ли два множества, объединение, пересечение, разность, симметрическая разность двух множеств.

16.6. Дробь.

Хранит числитель и знаменатель обыкновенной дроби. Методы: сократить дробь, сумма, разность, произведение, деление двух дробей, равны ли две дроби, правильная ли дробь.

16.7. Матрицы.

Хранит количество строк и столбцов, элементы целочисленной матрицы. Методы: сумма, разность, произведение двух матриц, умножение матрицы на скаляр, транспонирование матрицы, равны ли две матрицы, квадратная ли матрица, для квадратной матрицы – проверка симметричности относительно главной (побочной) диагонали.

16.8. Многочлены.

Хранит список одночленов (коэффициент, степень, степень, ...). Методы: принадлежит ли одночлен многочлену, сумма, разность, произведение, частное, остаток от деления двух многочленов, приведение подобных, равенство двух многочленов.

16.9 Треугольник.

Хранит координаты вершин треугольника на плоскости. Методы: равны ли два треугольника, площадь, периметр, высоты, медианы, биссектрисы, радиус вписанной, радиус описанной окружности, тип треугольника (равносторонний,

равнобедренный, прямоугольный, остроугольный, тупоугольный), поворот на заданный угол относительно (одной из вершин, центра описанной окружности).

16.10.Интервалы.

Хранит левую и правую границы интервала. Методы: длина интервала, смещение интервала (влево, вправо), сжатие (растяжение) интервала на заданный коэффициент, сравнение двух интервалов, сумма, разность двух интервалов.

16.11.Последовательность.

Хранит последовательность целых чисел. Методы: тип (убывающая, возрастающая, неубывающая, невозрастающая, геометрическая, арифметическая прогрессия), принадлежность элемента, равны ли две последовательности, максимум, минимум, установить разделители подпоследовательностей (положительные, отрицательные, нули, лок. максимумы, лок. минимумы, лок. экстремумы), наибольшая (наименьшая) подпоследовательность.

16.12.Предложение.

Хранит строку переменной длины представляющую собой предложение. Методы: добавить слово, удалить слово, вставить слово, количество букв, количество слов, самое длинное слово, самое короткое слово, есть ли в предложении заданное слово, слово под заданным номером, равны ли два предложения.

16.13.Текст.

Хранит строку переменной длины представляющую собой последовательность предложений. Методы: добавить предложение, удалить предложение, вставить предложение, количество (букв, слов, предложений), предложение под заданным номером, самое длинное предложение, самое короткое предложение, есть ли в тексте заданное предложение, равны ли два текста.

16.14.Расписание.

Хранит список (время начала, время окончания, место, текст (комментарий)). Методы: по дате – подсписок, очистить, добавить, заменить, удалить, вставить, по месту – время начала и длительность, можно ли вставить новый пункт, пересекаются ли два расписания.

16.15.Профиль местности.

Хранит последовательность высот вычисленных через равные промежутки по горизонтали. Методы: наибольшая высота, наименьшая высота, перепад высот (наибольший, суммарный), крутизна (наибольшая, средняя), сравнение двух профилей одинаковой длины (по перепаду, по крутизне).

17. Динамические структуры данных.

- 17.1. Используя стек преобразовать арифметическое выражение из инфиксной формы записи в постфиксную.
- 17.2. Используя стек преобразовать логическое выражение из инфиксной формы записи в постфиксную.
- 17.3. Преобразовать арифметическое выражение из постфиксной формы записи в инфиксную используя только необходимые скобки.
- 17.4. Преобразовать логическое выражение из постфиксной формы записи в инфиксную используя только необходимые скобки.
- 17.5. Используя стек вычислить значение арифметического выражения заданного в постфиксной форме записи.
- 17.6. Используя стек вычислить значение логического выражения заданного в постфиксной форме записи.
- 17.7. Упорядочить линейный список.
- 17.8. Подсчитать количество различных компонентов в списке.
- 17.9. Даны два линейных упорядоченных списка. Построить список объединяющий исходные с сохранением порядка как в первом списке.
- 17.10. Даны два линейных упорядоченных списка. Построить список объединяющий исходные с сохранением порядка как в более длинном списке.
- 17.11. Даны два линейных упорядоченных списка. Построить список объединяющий исходные с сохранением порядка как в более коротком списке.
- 17.12. Даны два линейных списка. Выяснить является ли второй список подсписком первого.
- 17.13. Даны два линейных списка. Выяснить является ли более короткий список подсписком более длинного.
- 17.14. Пусть два множества заданы в виде линейных списков. Выяснить является ли второе множество подмножеством первого.
- 17.15. Пусть два множества заданы в виде линейных списков. Построить линейный список представляющий пересечение двух множеств.
- 17.16. Пусть два множества заданы в виде линейных списков. Построить линейный список представляющий объединение двух множеств.
- 17.17. Пусть два множества заданы в виде линейных списков. Построить линейный список представляющий разность первого и второго множеств.
- 17.18. Пусть два множества заданы в виде линейных списков. Выяснить, равны ли два множества.

17.19. Пусть два множества заданы в виде линейных списков. Построить линейный список представляющий симметрическую разность двух заданных множеств.

17.20. Пусть многочлен задан в виде линейного списка. Элементом списка является информация о коэффициенте и степенях одночлена $kx^a y^b z^c$ (k, a, b, c – целые числа, $a, b, c \geq 0$). Найти сумму двух многочленов.

17.21. Пусть многочлен задан в виде линейного списка. Элементом списка является информация о коэффициенте и степенях одночлена $kx^a y^b z^c$ (k, a, b, c – целые числа, $a, b, c \geq 0$). Найти разность двух многочленов.

17.22. Пусть многочлен задан в виде линейного списка. Элементом списка является информация о коэффициенте и степенях одночлена $kx^a y^b z^c$ (k, a, b, c – целые числа, $a, b, c \geq 0$). Найти произведение двух многочленов.

17.23. Пусть многочлен задан в виде линейного списка. Элементом списка является информация о коэффициенте и степенях одночлена $kx^a y^b z^c$ (k, a, b, c – целые числа, $a, b, c \geq 0$). Найти частное от деления двух многочленов.

17.24. Пусть многочлен задан в виде линейного списка. Элементом списка является информация о коэффициенте и степенях одночлена $kx^a y^b z^c$ (k, a, b, c – целые числа, $a, b, c \geq 0$). Найти остаток от деления одного многочлена на другой.

17.25. Пусть многочлен задан в виде линейного списка. Элементом списка является информация о коэффициенте и степенях одночлена $kx^a y^b z^c$ (k, a, b, c – целые числа, $a, b, c \geq 0$). Найти частное и остаток от деления одного многочлена на другой.

17.26. Пусть многочлен задан в виде линейного списка. Элементом списка является информация о коэффициенте и степенях одночлена $kx^a y^b z^c$ (k, a, b, c – целые числа, $a, b, c \geq 0$). Привести подобные члены в многочлене.

17.27. Пусть многочлен задан в виде линейного списка. Элементом списка является информация о коэффициенте и степени одночлена kx^a (k, a – целые числа, $a \geq 0$). Найти производную.

17.28. Преобразовать список таким образом, чтобы каждый компонент присутствовал бы только один раз.

17.29. Удалить из списка те компоненты, которые присутствуют более одного раза.

17.30. Даны два линейных списка (поле информации – целое число). Оставить в первом списке только те компоненты, которые отличаются не более чем на k от какого-нибудь из компонентов второго списка.

17.31. Даны два линейных списка (поле информации – целое число). Оставить в первом списке только те компоненты, которые отличаются не менее чем на k от какого-нибудь из компонентов второго списка.

- 17.32. Пусть последовательность целых чисел задана в виде линейного списка. Найти самую длинную возрастающую подпоследовательность.
- 17.33. Дан линейный список целых чисел. Построить по заданному списку два: список отрицательных и список неотрицательных чисел с сохранением порядка их следования в исходном списке.
- 17.34. Дан линейный список целых чисел. Удалить все компоненты расположенные между наибольшим и наименьшим компонентами.
- 17.35. Дан линейный список целых чисел. Удалить все компоненты списка кроме локальных экстремумов.
- 17.36. Дан линейный символьный список. Список сбалансирован по круглым скобкам (т.е. для каждой открывающей скобки есть соответствующая закрывающая), скобки могут быть вложенными. Удалить из списка все символы, расположенные между самыми внутренними скобками.
- 17.37. Дан линейный список целых чисел. Удалить из списка те компоненты, значения которых отличаются от соседей больше чем в среднем по списку.
- 17.38. Дан линейный список целых чисел. Удалить из списка те компоненты, значения которых отличаются от соседей меньше чем в среднем по списку.
- 17.39. Пусть разреженные матрицы заданы в виде списочной структуры. Найти сумму двух матриц.
- 17.40. Пусть разреженные матрицы заданы в виде списочной структуры. Найти произведение двух матриц.
- 17.41. Пусть разреженная квадратная матрица задана в виде списочной структуры. Транспонировать матрицу.
- 17.42. Пусть бинарные деревья заданы в виде связного списка. Вывести все листья дерева.
- 17.43. Пусть бинарные деревья заданы в виде связного списка. Вывести все узлы дерева по уровням.
- 17.44. Пусть бинарные деревья заданы в виде связного списка. Вывести все узлы наиболее длинной ветви дерева.
- 17.45. Пусть бинарные деревья заданы в виде связного списка. Проверить, равны ли два дерева.
- 17.46. Пусть бинарные деревья заданы в виде связного списка. Проверить, подобны ли два дерева.
- 17.47. Пусть бинарные деревья заданы в виде связного списка. Проверить является ли одно дерево поддеревом другого.
- 17.48. Пусть арифметическое выражение задано в виде бинарного дерева. Вычислить значение этого выражения.

Примеры на языке Java.

18. Вычисление значения функции как суммы бесконечного сходящегося ряда.

Написать программу нахождения суммы ряда:

$$1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} + \dots$$

с заданной точностью ϵ . Использовать рекуррентные соотношения при вычислении очередного члена ряда. Вычисление заканчивается, если модуль очередного слагаемого становится меньше заданного значения точности (ϵ), причем для этого ряда (при $|x| < 1$) абсолютная величина суммы всех отброшенных членов ряда будет меньше ϵ . Для оценки правильности расчетов предусмотреть вычисление по контрольной формуле: e^x .

```
package infiniteseries;
import java.io.*;
public class InfiniteSeries{
    public static void main(String[] args)throws IOException{
        double x, p, n, sum, f, eps;
        BufferedReader br=new BufferedReader(new
            InputStreamReader(System.in, "CP866"));
        System.out.print("Введите x(0<|x|<1): ");
        x=Double.parseDouble(br.readLine());
        System.out.print("Введите точность(eps): ");
        eps=Double.parseDouble(br.readLine());
        f=Math.exp(x);
        p=1;
        sum=p;
        n=1;
        while(Math.abs(p)>=eps){
            p *= x/n;
            sum += p;
            n++;
        }
        System.out.println("x="+x+"\neps="+eps+"\nf="+
            f+"\nsum="+sum);
    }
}
```

19. Замена в строке одной подстроки на другую.

Пусть предложение задано в виде строки символов. Слова в предложении отделяются друг от друга одним пробелом. Написать программу замены в предложении одного заданного слова другим.

```
package wordexchange;
```

```

import java.io.*;
import java.util.*;
public class WordExchange{
    public static void main(String[] args)throws IOException{
        BufferedReader br=new BufferedReader(new
            InputStreamReader(System.in, "CP866"));
        String s, w1, w2;
        int i=0;
        System.out.print("Введите предложение: ");
        s=br.readLine();
        System.out.print("Введите слово для поиска:");
        w1=br.readLine();
        System.out.print("Введите слово для замены:");
        w2=br.readLine();
//первый вариант
        i=s.indexOf(w1); //Находим номер символа в строке s
                        //с которого подстрока w1 //входит
                        //в строку s. (-1 если такой нет).
        if(i != -1){
            s=s.substring(0,i)+w2+s.substring(i+w1.length());
        }
//второй вариант
        s=s.replaceFirst(w1,w2);
//
        System.out.println("В результате замены получили: "+s);
    }
}

```

20. Поиск в строке всех вхождений заданной подстроки.

Найти номера символов в заданной строке s, с которых начинаются все подстроки равные строке w.

```

package allsubstring;
import java.io.*;
import java.util.*;
public class AllSubString{
    public static void main(String[] args)throws IOException{
        BufferedReader br=new BufferedReader(new
            InputStreamReader(System.in, "CP866"));
        String s, w;
        int i, k=0;
        System.out.print("Введите предложение: ");
        s=br.readLine();
        System.out.print("Введите слово для поиска: ");
        w=br.readLine();
        do{
            i=s.indexOf(w,k);
            k=i+1;
            if(i != -1)

```

```

        System.out.print(i+"\t");
    }while(i != -1);
    System.out.println();
}
}

```

21. В строке поменять местами две подстроки.

В предложении поменять местами самое длинное и самое короткое слова.

Первый вариант

```

//Перебираем все символы предложения по одному
//Как только очередной символ - пробел,
// значит готово очередное слово
package wordsplacesexchange;
import java.io.*;
import java.util.*;
public class WordsPlacesExchange{
    public static void main(String[] args)throws IOException{
        BufferedReader br=new BufferedReader(new
            InputStreamReader(System.in, "CP866"));
        String s;
        int i;
        System.out.print("Введите предложение: ");
        s=br.readLine();
        String wmin=s, wmax="", w="";
        char c;
        s=s.trim()+" ";
        int imax=0, imin=0;
        for(i=0; i<s.length; i++){
            c=s.charAt(i);
            if(c != ' '){
                w=w+Character.toString(c);
            }else{
                if(wmin.length() > w.length()){
                    wmin=w;
                    imin=i-w.length();
                }
                if(wmax.length() < w.length()){
                    wmax=w;
                    imax=i-w.length();
                }
                w="";
            }
        }
        if(imax > imin){
            s=s.substring(0, imin)+wmax+
                s.substring(imin+wmin.length(), imax)+
                wmin+s.substring(imax+wmax.length());
        }
    }
}

```

```

        if(imax < imin){
            s=s.substring(0,imax)+wmin+
              s.substring(imax+wmax.length(),imin)+
              wmax+s.substring(imin+wmin.length());
        }
        System.out.println("В результате замены получили: "+s);
    }
}

```

Второй вариант

```

//Используем класс StringTokenizer,
// который позволяет получить очередное слово - лексему(token)
package wordsplacesexchange;
import java.io.*;
import java.util.*;
public class WordsPlacesExchange{
    public static void main(String[] args)throws IOException{
        BufferedReader br=new BufferedReader(new
            InputStreamReader(System.in, "CP866"));
        String s;
        int i;
        System.out.print("Введите предложение: ");
        s=br.readLine();
        String wmin=s, wmax="", w;
        StringTokenizer st=StringTokenizer(s, " .,-!-?;: ");
        while(st.hasMoreTokens()){ //Если есть еще слова
            w=st.nextToken(); //получаем очередное слово
            if(wmin.length()>w.length()) wmin=w;
            if(wmax.length()<w.length()) wmax=w;
        }
        imax=s.indexOf(wmax);
        imin=s.indexOf(wmin);
        if(imax > imin){
            s=s.substring(0,imin)+wmax+
              s.substring(imin+wmin.length(),imax)+
              wmin+s.substring(imax+wmax.length());
        }
        if(imax < imin){
            s=s.substring(0,imax)+wmin+
              s.substring(imax+wmax.length(),imin)+
              wmax+s.substring(imin+wmin.length());
        }
        System.out.println("В результате замены получили: "+s);
    }
}

```

Третий вариант

```

//Используем метод split класса String,
// который возвращает массив строк - слов (лексем)

```

```

package wordsplacesexchange;
import java.io.*;
import java.util.*;
public class WordsPlacesExchange{
    public static void main(String[] args)throws IOException{
        BufferedReader br=new BufferedReader(new
            InputStreamReader(System.in, "CP866"));
        String s;
        int i;
        System.out.print("Введите предложение: ");
        s=br.readLine();
        String wmin=s, wmax="";
        String[] tokens=s.split("");
        for(i=0; i<tokens.length; i++){
            if(wmin.length()>tokens[i].length()){
                wmin=tokens[i];
                imin=i;
            }
            if(wmax.length()<tokens[i].length()){
                wmax=tokens[i];
                imax=i;
            }
        }
        s="";
        for(i=0; i<tokens.length; i++){
            if(i==imax)
                s=s+" "+wmin;
            else
                if(i==imin)
                    s=s+" "+wmax;
                else
                    s=s+" "+tokens[i];
        }
        System.out.println("В результате замены получили: "+s);
    }
}

```

22. Пример функции, возвращающей одномерный массив

Инвертировать компоненты одномерного массива целых чисел.

```

package arrayreturn;
import java.io.*;

public class ArrayReturn{
    public static int[] reversArray(int[] a){
        int[] b=new int[a.length];
        for(int i=0; i<b.length; i++){
            b[i]=a[b.length-i-1];
        }
        return b;
    }
}

```

```

}

public static void main(String[] args) throws IOException{
    int[] array;
    int n;
    BufferedReader br=new BufferedReader(new
        InputStreamReader(System.in));
    System.out.print("Введите эл-ты массива через пробел");
    String s=br.readLine();
    String[] sa= s.split(" ");
    n=sa.length;
    array=new int[n];
    for(int i=0; i<n; i++){
        array[i]=Integer.parseInt(sa[i]);
    }
    array=reversArray(array);
    for(int i=0; i<n; i++){
        System.out.print(array[i]+ " ");
    }
    System.out.println();
}
}

```

23. Пример функции, возвращающей двумерный массив

Даны два одномерных массива целых чисел одинаковой длины. Рассматривая первый массив как вектор-столбец, а второй как вектор-строку вывести матрицу произведения вектора-столбца на вектор-строку.

```

package columnrowmatrix;
import java.io.*;

public class ColumnRowMatrix{

    public static int[][] colRowMatrix(int[] col, int[] row){
        int[][] matrix;
        if(col.length!= row.length) return null;
        int n=col.length;
        matrix=new int[n][n];
        for(int i=0; i<n; i++)
            for(int j=0; j<n; j++)
                matrix[i][j]=col[i]*row[j];
        return matrix;
    }

    public static void main(String[] args) throws IOException{
        int[] a, b;
        int[][] c;
        BufferedReader br=new BufferedReader(new
            InputStreamReader(System.in));
    }
}

```

```

System.out.print("Введите первый массив через пробел");
String s=br.readLine();
String[] as= s.split(" ");
n=as.length;
a=new int[n];
for(int i=0; i<n; i++){
    a[i]=Integer.parseInt(as[i]);
}
System.out.print("Введите второй массив через пробел");
String s=br.readLine();
String[] as= s.split(" ");
int m=as.length;
if(n != m){
    System.out.println("Длины массивов не равны!");
    return;
}
b=new int[m];
for(int i=0; i<m; i++){
    b[i]=Integer.parseInt(as[i]);
}
c=colRowMatrix(a, b);
for(int i=0; i<n; i++){
    for(int j=0; j<n; j++)
        System.out.print(c[i][j]+ "\t");
    System.out.println();
}
}
}

```

24. Пример функции, возвращающей строку

Написать метод преобразующий одномерный массив целых чисел в строку.

```

package arraytostring;
import java.io.*;

public class ArrayToString{

    public static String vectorToString(int[] vector){
        String s = "";
        int n = vector.length;
        for(int i=0; i<n; i++)
            s += Integer.toString(vector[i]) + " ";
        return s;
    }

    public static void main(String[] args)throws IOException{
        int[] a;
        int n;
        BufferedReader br=new BufferedReader(new
            InputStreamReader(System.in));
    }
}

```

```

        System.out.print("Введите эл-ты массива через пробел");
        String s=br.readLine();
        String[] as= s.split(" ");
        n=as.length;
        a=new int[n];
        for(int i=0; i<n; i++){
            a[i]=Integer.parseInt(as[i]);
        }
        System.out.println("В результате получили:");
        System.out.println(vectorToString(a));
    }
}

```

25. Пример рекурсивной функции вычисляющей факториал

```

package recursivefactorial;
import java.io.*;

public class RecursiveFactorial{

    public static int factorial(int n){
        if(n <= 0)
            return 1;
        else
            return factorial(n-1)*n;
    }

    public static void main(String[] args)throws IOException{
        BufferedReader br=new BufferedReader(new
            InputStreamReader(System.in));
        int x, f;
        System.out.print("Введите целое число >= 0");
        x = Integer.parseInt(br.readLine());
        f = factorial(x);
        System.out.println("Факториал " + x + " равен " + f);
    }
}

```

26. Пример рекурсивной функции вычисляющей максимум в одномерном массиве

```

package recursivefunctionmax;
import java.io.*;

public class RecursiveFunctionMax{

    public static int maxInVector(int[] vector, int n){
        int m;
        if(n <= 0)
            return vector[0];
    }
}

```

```

        else{
            m = maxInVector(vector, n-1);
            if(m < vector[n])
                return vector[n];
            else
                return m;
        }
    }

public static void main(String[] args) throws IOException{
    int[] a;
    int n;
    BufferedReader br=new BufferedReader(new
        InputStreamReader(System.in));
    System.out.print("Введите эл-ты массива через пробел");
    String s=br.readLine();
    String[] as= s.split(" ");
    n=as.length;
    a=new int[n];
    for(int i=0; i<n; i++){
        a[i]=Integer.parseInt(as[i]);
    }
    System.out.println("max = " + maxInVector(a, n-1));
}
}

```

27. Пример рекурсивной процедуры вычисляющей максимум в одномерном массиве

```

package recursiveproceduremax;
import java.io.*;

public class RecursiveProcedureMax{

    public static void maxInArray(int[] vector, int n, ObjInt m){
        if(n <= 0)
            m.i = vector[0];
        else{
            maxInArray(vector, n-1, m);
            if(m.i < vector[n]) m.i = vector[n];
        }
    }

    public static void main(String[] args) throws IOException{
        int[] a;
        int n;
        ObjInt x = new ObjInt();
        BufferedReader br=new BufferedReader(new
            InputStreamReader(System.in));
        System.out.print("Введите эл-ты массива через пробел");
    }
}

```

```

        String s=br.readLine();
        String[] as= s.split(" ");
        n=as.length;
        a=new int[n];
        for(int i=0; i<n; i++){
            a[i]=Integer.parseInt(as[i]);
        }
        maxInArray(a, n-1, x);
        System.out.println("max = " + x.i);
    }
}

class ObjInt{
    public int I;
}

```

28. Поиск подпоследовательности в последовательности

В последовательности целых чисел подпоследовательности разделяются локальными максимумами. Найти подпоследовательность наибольшей длины.

```

package subsequence;
import java.io.*;

public class SubSequence {

    public static void main(String[] args)throws IOException {
        int[] a, b;
        int n, k=0;
        BufferedReader br = new BufferedReader(new
            InputStreamReader(System.in));
        System.out.print("Введите последовательность "+
            " целых чисел через пробел\n>");
        String s = br.readLine();
        String[] sa = s.split(" ");
        n = sa.length;
        a = new int[n];
        for(int i=0; i<n; i++)
            a[i] = Integer.parseInt(sa[i]);
        b = new int[n];
        if(a[0] > a[1]){
            b[k] = 0;
            k++;
        }
        for(int i=1; i<n-1; i++)
            if(a[i] > a[i-1] && a[i] > a[i+1]){
                b[k] = i;
                k++;
            }
        if(a[n-1] > a[n-2]){

```

```

        b[k] = n-1;
        k++;
    }

    System.out.println("Порядковые номера разделителей:");
    for(int i=0; i<k; i++)
        System.out.print(b[i] + " ");
    System.out.println();

    int curbeg, curlen, maxbeg, maxlen;
    if(b[0] != 0){
        maxbeg = 0;
        maxlen = b[0];
    }else{
        maxbeg = b[0]+1;
        maxlen = b[1]-b[0]-1;
    }
    for(int i=1; i<k-1; i++){
        curbeg = b[i]+1;
        curlen = b[i+1]-curbeg;
        if(maxlen < curlen){
            maxbeg = curbeg;
            maxlen = curlen;
        }
    }
    if(b[k-1] != n-1){
        curbeg = b[k-1]+1;
        curlen = n-1-curbeg;
        if(maxlen < curlen){
            maxbeg = curbeg;
            maxlen = curlen;
        }
    }
    System.out.println("Подпоследовательность " +
        " наибольшей длины:");
    for(int i=0; i<maxlen; i++)
        System.out.print(a[maxbeg+i] + " ");
    System.out.println();
}
}

```