

Государственное бюджетное образовательное учреждение
высшего профессионального образования города Москвы
«Московский городской педагогический университет»
Самарский филиал

Г. А. КЛЕКОВКИН
П. С. БОГДАНОВ

РЕШЕНИЕ КОМБИНАТОРНЫХ ЗАДАЧ С ИСПОЛЬЗОВАНИЕМ СРЕДЫ MAXIMA

Учебное пособие

Самара
2015

УДК 519.1
ББК 22.141
К48

Печатается по решению
Ученого совета СФ ГБОУ ВПО МГПУ

Рецензенты:

доктор педагогических наук, профессор кафедры фундаментальной
и компьютерной математики Вятского государственного
гуманитарного университета *С. И. Калинин*;

доктор физико-математических наук, профессор,
заведующий кафедрой алгебры, геометрии и методики
обучения математике Стерлитамакского филиала Башкирского
государственного университета *П. Н. Михайлов*

Клековкин Г. А., Богданов П. С.

К48 Решение комбинаторных задач с использованием
среды MAXIMA: учебное пособие / Г. А. Клековкин,
П. С. Богданов. – Самара: СФ ГБОУ ВПО МГПУ, 2015. –
88 с.

ISBN 978-5-9905895-3-7

В пособии рассмотрены примеры использования системы
компьютерной математики Maxima при решении комбинатор-
ных задач на выбор и упорядочивание элементов конечного
множества, разбиение конечных множеств и разложение нату-
ральных чисел на слагаемые. Даны инструкции и индивиду-
альные задания для трех лабораторных работ.

Предназначено для студентов вузов, обучающихся по на-
правлению 050100 Педагогическое образование. Может быть
использовано при проведении лабораторных работ в курсе
«Дискретная математика», а также при разработке и проведе-
нии специальных курсов.

УДК 519.1
ББК 22.141

ISBN 978-5-9905895-3-7

© СФ ГБОУ ВПО МГПУ, 2015
© Г. А. Клековкин, П. С. Богданов, 2015

СОДЕРЖАНИЕ

Введение	4
Основы работы в системе Maxima	6
Лабораторная работа № 1 «Вычисление комбинаторных чисел, заданных формулами»	21
Лабораторная работа № 2 «Рекуррентные соотношения» ...	36
Лабораторная работа № 3 «Производящие функции»	59
Литература	87

ВВЕДЕНИЕ

Учебно-методическое пособие предназначено для организации и проведения лабораторных занятий по теме «Перечислительная комбинаторика» в курсе дискретной математики.

Существует три основных способа точного вычисления комбинаторных чисел. Первый способ используется тогда, когда перечисляющую функцию удастся найти в явном виде, т. е. записать в виде выражения, составленного из известных функций с помощью конечного числа основных арифметических операций. Второй способ задания и вычисления комбинаторных чисел основан на понятии рекуррентного соотношения. Применяя этот способ, находят одно или несколько первых комбинаторных чисел требуемого вида и указывают правило (рекуррентное соотношение), позволяющее вычислить по ним последующие числа. При третьем способе комбинаторные числа ищутся методом производящих функций. Суть этого метода состоит в сопоставлении последовательности искомых комбинаторных чисел формального степенного ряда (производящей функции), коэффициенты которого являются членами этой последовательности.

К сожалению, решение многих численных комбинаторных задач, как чисто учебных, так и прикладных, с помощью любого из перечисленных способов может быть сопряжено с громоздкими вычислениями. Поэтому на практических занятиях решение задач нередко заканчивается записью нужных для получения численного результата формул, рекуррентных соотношений или производящих функций. В лучшем случае преподаватель на аудиторных занятиях совместно со студентами доводит «с помощью бумаги и ручки» до численного результата две-три задачи. Вместе с тем, практика показывает, что некоторые студенты остаются неудовлетворенными такими решениями. Поэтому данное пособие адресовано в первую очередь именно тем студентам, которые привыкли заканчивать решение точными вычислениями.

Новые возможности для автоматизации численных и символьных вычислений открывают системы компьютерной математики. Существует множество коммерческих и свободно распространяемых систем компьютерной математики общего назначения. До недавнего времени наибольшей популярностью у нас в стране пользовались коммерческие системы Maple, Mathematica и MathCad. Однако из-за высокой стоимости эти программы нашим индивидуальным пользователям практически недоступны. Поэтому среди них растет интерес к свободно распространяемым программам, прежде всего к системе Maxima.

Система Maxima обладает весьма широкими возможностями для численных и символьных вычислений в различных областях математики

и во многих отношениях не уступает таким коммерческим системам, как Maple или Matematica. При этом система имеет небольшой размер дистрибутива, легко устанавливается, существуют ее русифицированные графические интерфейсы. Еще одним важным достоинством системы Maxima является ее многоплатформенность, она доступна как пользователям операционных систем Linux, так и пользователям Windows. Наконец, она чрезвычайно проста для освоения.

Система Maxima, таким образом, может стать идеальным объектом изучения и инструментом численных и символьных вычислений для школьников старших классов и студентов вузов, ее также могут использовать в своей работе как профессиональные математики, так и специалисты, имеющие дело с приложениями математики, для проведения сложных расчетов и исследований.

В стране постепенно формируется банк учебной и методической литературы, необходимой для массового внедрения системы Maxima в школьное и вузовское преподавание математики. Почти все имеющиеся пособия похожи по структуре и содержанию, авторы дают описание возможностей системы, ее интерфейса, правил ввода в систему команд и различных типов данных, рассматривают примеры решения задач, главным образом из базового курса высшей математики.

В данном пособии даны примеры использования системы Maxima при решении задач пересчетной комбинаторики. Авторы пособия при этом предполагают, что читатель уже знаком с интерфейсом системы и имеет начальные навыки работы с ней, полученные в базовом курсе математики. Поэтому общие сведения о работе в системе Maxima даны в пособии в минимально необходимом объеме. Для уточнения и расширения этих сведений читатель всегда может обратиться либо к англоязычному справочнику системы, либо к существующим руководствам по работе с ней, которые имеются в сети Интернет. (Отметим прежде всего учебно-методические пособия Н. А. Стахина [3] и Е. А. Чичкарёва [4].)

В пособии приведены инструкции и индивидуальные задания к трем лабораторным работам, которые могут быть проведены в практикуме по дискретной математике либо в факультативном курсе по системам компьютерной математики: «Вычисление комбинаторных чисел, заданных формулами», «Рекуррентные соотношения» и «Производящие функции».

Используемая в пособии терминология и обозначения комбинаторных чисел даны в полном соответствии с учебным пособием Г. А. Клековкина [2]. При подборке задач для индивидуальных заданий главным образом использовалась книга Н. Я. Виленкина [1].

ОСНОВЫ РАБОТЫ В СИСТЕМЕ MAXIMA

Как было отмечено во введении, авторы предполагают, что читатель уже имел опыт выполнения численных и символьных вычислений в среде Maxima. Поэтому в данном пункте лишь кратко напоминаются некоторые правила работы в этой среде и описываются функции, с которыми наиболее часто приходится сталкиваться при проведении вычислений в комбинаторных задачах, традиционно входящих в курс дискретной математики. Описания даны применительно к графическому интерфейсу wxMaxima, существенно облегчающему работу при решении типовых математических задач.

Ввод команд и вывод результатов. После запуска Maxima появляется окно программы, в рабочем поле которого вводятся и записываются все команды, обрабатываемые системой. Команда в Maxima – это любая комбинация выражений и встроенных функций. Разделителем команд является символ «;» (точка с запятой). После ввода команды для ее обработки и вывода результата необходимо нажать клавиши <Ctrl>+<Enter>.

Пример:

```
(%i1) 1/2+2/3;  
(%o1)  $\frac{7}{6}$ 
```

Предусмотрена возможность ввода сразу нескольких команд в одной строке, для этого одна команда от другой отделяется символом «;». При этом формируется одна строка ввода и столько строк вывода, сколько команд было задано.

Пример:

```
(%i2) 1/2+2/3;3*4-2;sqrt(9);  
(%o2)  $\frac{7}{6}$   
(%o3) 10  
(%o4) 3
```

В рассмотренных примерах система в ответе вывела значение выражения, хотя, казалось бы, мы не задали никакой команды. Это объясняется тем, что в системе Maxima по умолчанию является активной функция упрощения: если вводимое выражение можно упростить, то система обязательно это делает.

Для обозначения конца ввода команды вместо точки с запятой можно использовать знак \$, позволяющий не выводить на экран результат вычисления (команда при этом будет исполняться). Это удобно тогда, когда визуализация результата не нужна, например, при загрузке файлов с помощью функции **load**.

В том случае, когда выражение надо отобразить, а не вычислить, перед ним ставится знак «'» (одинарная кавычка). Но этот прием не работает, когда выражение имеет явное численное значение.

Примеры:

```
(%i6) '(x^2-4)/(x-2);
```

```
(%o6) 
$$\frac{x^2-4}{x-2}$$

```

```
(%i7) 'cos(%pi);
```

```
(%o7) -1
```

Две одинарных кавычки, последовательно примененные к выражению во входной строке, приводят к замещению входной строки результатом вычисления вводимого выражения.

Из рассмотренных примеров видно, что после ввода каждой команде присваивается ее порядковый номер: (%i1), (%i2), Соответствующие порядковые номера присваиваются и результатам вычислений: (%o1), (%o2), Это позволяет при дальнейшей записи команд ссылаться на ранее записанные команды и результаты вычислений.

Примеры:

```
(%i10) (x^2-4)/(x-2);
```

```
(%o10) 
$$\frac{x^2-4}{x-2}$$

```

```
(%i11) (x^2-4)/(x+2);
```

```
(%o11) 
$$\frac{x^2-4}{x+2}$$

```

```
(%i12) (%i10)*(%i11);
```

```
(%o12) 
$$\frac{(x^2-4)^2}{(x-2)(x+2)}$$

```

```
(%i13) (%o10)*(%o11);
```

```
(%o13) 
$$\frac{(x^2 - 4)^2}{(x - 2)(x + 2)}$$

```

Операторы. Для обозначения арифметических операций в Махима используются знаки: «+» – сложение, «-» – вычитание, «*» – умножение, «/» – деление. Возведение в степень можно обозначать тремя способами: $^$, $^{^}$, ** . Извлечение корня степени n записывают, как степень $^{(1/n)}$.

Для обозначения приоритета операции, как обычно, при записи команд используются круглые скобки ().

Операторы сравнения: < (меньше), <= (меньше, равно), > (больше), >= (больше, равно).

Логические операторы: **and** (и), **or** (или), **not** (не).

Константы. В Махима для удобства вычислений имеется ряд встроенных констант. При выполнении лабораторных работ будут использоваться константы:

Название	Обозначение
π (число пи)	%pi
e (экспонента)	%e
$i = \sqrt{-1}$ (мнимая единица)	%i
Истина	true
Ложь	false

Числа. Все функции и операторы Махима работают как с действительными, так и с комплексными числами. При записи действительных чисел числитель и знаменатель обыкновенных дробей разделяется при помощи символа «/», целая и дробная части десятичных дробей разделяются символом точка, перед отрицательными числами ставится знак минус. Комплексные числа записываются в алгебраической форме, т. е. в виде $a + b * \%i$, где a и b – соответственно действительная и мнимая части числа.

Переменные. Для выполнения промежуточных расчетов и хранения их результатов применяются переменные, которым могут присваиваться определенные значения. В виде значения такой переменной может выступать любое математическое выражение. При вводе названий переменных важен регистр букв, так как пе-

переменные x и X – это две разные переменные (сказанное в равной мере относится к записи функций и констант). Присваивание значения переменной осуществляется с использованием оператора «:» (двоеточие).

Примеры:

```
(%i1) a:2;
(%o1) 2
(%i2) b:x^2+3*x;
(%o2) x^2+3 x
```

В первом примере пользователем было введено $a:2$; – то есть переменной a присваивалось значение 2; (%i1) означает, что в данной строке и далее предполагается пользовательский ввод. Второй пример показывает, что переменной можно присваивать значение другой переменной или функции.

Математические функции. В Maxima имеется набор встроенных математических функций. Для ввода математической функции записывается ее название, а затем в круглых скобках через запятую перечисляются аргументы. Напомним, что некоторые обозначения функций, принятые в Maxima, отличаются от обозначений, используемых в отечественной математической литературе. При выполнении лабораторных работ будут использоваться функции:

Название	Обозначение
Квадратный корень (\sqrt{x})	$\text{sqrt}(x)$
Экспонента (e^x)	$\text{exp}(x)$
Натуральный логарифм ($\ln x$)	$\text{log}(x)$
Синус ($\sin x$)	$\text{sin}(x)$
Косинус ($\cos x$)	$\text{cos}(x)$
Абсолютная величина ($ x $)	$\text{abs}(x)$
Знак числа $\left(\text{sign} x = \begin{cases} 1, & \text{если } x > 0, \\ 0, & \text{если } x = 0, \\ -1, & \text{если } x < 0 \end{cases} \right)$	$\text{signum}(x)$

Пример:

```
(%i3) sqrt(x);log(x);  
(%o3)  $\sqrt{x}$   
(%o4)  $\log(x)$ 
```

Пользовательские функции. Пользователь может создавать собственные функции с помощью оператора задания функции пользователя «:=» (двоеточие и равно). Для этого сначала указывается название функции и в скобках перечисляются названия аргументов, затем после знаков «:=» следует описание функции. После задания пользовательская функция вызывается точно так же, как и встроенные функции Maxima. Это значительно облегчает работу с ней, поскольку к ней можно обращаться по имени и легко и удобно вычислять значения в заданных точках.

Пример:

```
(%i5) f(x):=x^3+2*x-1;  
(%o5)  $f(x) := x^3 + 2x - 1$   
(%i6) x:3;f(x);  
(%o6) 3  
(%o7) 32
```

Любое имя можно очистить от присвоенного ему выражения функцией **kill()** и тем самым освободить занимаемую этим выражением память. Для этого нужно просто набрать **kill(name)**, где name – имя уничтожаемого выражения; причем это может быть как имя, назначенное пользователем, так и любая ячейка ввода или вывода. Точно так же можно очистить разом всю память и освободить все имена, введя **kill(all)**. В этом случае очистятся и все ячейки ввода-вывода, а их нумерация опять начнется с единицы.

Списки. Списки – базовые строительные блоки в Maxima, представляющие собой конечные упорядоченные последовательности элементов. Элементами списка могут быть числа, переменные, выражения, другие списки; в частности, список может содержать один элемент, быть пустым. Для задания списка нужно в квадратных скобках через запятую перечислить его элементы.

Примеры:

```
(%i1) L1: [0, 1, 2, 3, 4];
(%o1) [0, 1, 2, 3, 4]
(%i2) L2: [x, y, z, x+y, x+y+z];
(%o2) [x, y, z, y+x, z+y+x]
(%i3) L3: [[1, 2], [1, 3], [2, 3], [1, 2, 3]];
(%o3) [[1, 2], [1, 3], [2, 3], [1, 2, 3]]
(%i4) L4: [];
(%o4) []
```

Если элементы списка могут быть заданы с помощью формулы, то для его формирования удобно использовать команду **makelist()**.

Пример:

```
(%i7) L5: makelist(k*(k+1)/2, k, 1, 10);
(%o7) [1, 3, 6, 10, 15, 21, 28, 36, 45, 55]
```

Здесь сформирован список с именем *L5*, состоящий из десяти элементов, значения которых находятся по формуле $\frac{k(k+1)}{2}$.

В программе *Maxima* имеется широкий набор функций для работы со списками и набор функций для работы с элементами списка. Их описание можно найти в [4], а также в справочной системе программы.

Массивы. Массивы – упорядоченные совокупности однородных объектов, помеченных индексами. Число индексов не должно превышать пяти.

Для создания массивов в *Maxima* существует функция **array** (*name*, *dim_1*, ..., *dim_n*) и ее модификации. Здесь *name* – имя переменной, в которой хранится массив, *dim_1*, ..., *dim_n* – размеры *n* – мерного массива ($n \leq 5$). Напомним также, что нумерация элементов массива начинается с 0 и заканчивается на *dim_i*, где $i = 1, \dots, n$.

Примеры:

```
(%i42) array(A, 9);
(%o42) A
(%i75) array(AA, 2, 2);
(%o75) AA
```

В первом примере сформирован одномерный массив A длиной 10, во втором – двумерный массив AA , имеющий 3 строки и 3 столбца.

Массив может быть заполнен путем непосредственного задания значений всех его элементов.

Примеры:

```
(%i43) A[0]:0;A[1]:2;A[2]:4;A[3]:6;A[4]:8;
      A[5]:10;A[6]:12;A[7]:14;A[8]:16;A[9]:18;
(%o43) 0
(%o44) 2
(%o45) 4
(%o46) 6
(%o47) 8
(%o48) 10
(%o49) 12
(%o50) 14
(%o51) 16
(%o52) 18

(%i76) AA[0,0]:1;AA[0,1]:0;AA[0,2]:0;
      AA[1,0]:0;AA[1,1]:2;AA[1,2]:0;
      AA[2,0]:0;AA[2,1]:0;AA[2,2]:3;
(%o76) 1
(%o77) 0
(%o78) 0
(%o79) 0
(%o80) 2
(%o81) 0
(%o82) 0
(%o83) 0
(%o84) 3
```

Вывод значений элементов массива, как видим, осуществляется в столбец. Функция **listarray(name)** позволяет преобразовать массив в список. Порядок включения элементов массива в список – по строкам.

Примеры:

```
(%i63) listarray(A);  
(%o63) {0, 2, 4, 6, 8, 10, 12, 14, 16, 18}  
(%i85) listarray(AA);  
(%o85) {1, 0, 0, 0, 2, 0, 0, 0, 3}
```

Для доступа к элементам массива используется порядковые номера элемента по каждой из размерностей, указанные в квадратных скобках через запятую.

Примеры:

```
(%i89) A[0];A[4];A[9];  
(%o89) 0  
(%o90) 8  
(%o91) 18  
(%i87) AA[1, 1];AA[1, 2];  
(%o87) 2  
(%o88) 0
```

Во многих случаях двумерные массивы удобнее задавать построчно в виде матрицы с помощью функции **matrix** (*row_1*, ..., *row_n*).

Пример:

```
(%i1) AA: (matrix([1, 0, 0], [0, 2, 0], [0, 0, 3]));  
(%o1) 
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

```

Следующий пример демонстрирует задание двумерного массива $B = \|b_{ij}\|$ размерности 4×4 , элементы которого заданы с помощью формулы $b_{ij} = i^2 - j^3$.

```
(%i2) B: genmatrix(lambda([i, j], i^2 - j^3), 4, 4);  
(%o2) 
$$\begin{bmatrix} 0 & -7 & -26 & -63 \\ 3 & -4 & -23 & -60 \\ 8 & 1 & -18 & -55 \\ 15 & 8 & -11 & -48 \end{bmatrix}$$

```

Программирование в Maxima. Во встроенном языке системы заложены основные исполнимые операторы, которые есть в любом языке программирования. Напомним их синтаксис.

Условный оператор в общем виде может быть записан следующим образом:

**if *cond_1* then *expr_1* else if *cond_2* then *expr_2* elseif ... else
expr_0;**

причем часть «else if *cond_2* then *expr_2* else if ...» можно опустить. Данный оператор выполняет действия *expr_1*, если условие *cond_1* верно, и так далее, то есть выполняет действие *expr_k*, если условие *cond_k* верно. Если ни одно из условий не верно, то выполняется действие *expr_0*. Кроме того, можно опустить и часть «else *expr_0*». В этом случае при невыполнении всех условий *cond_k* в качестве результата работы оператора получим «*false*».

Примеры:

1. Неполная форма условного оператора:

```
(%i25) b(a):=if a<0 then 5;
(%o25) b(a):=if a<0 then 5
(%i26) b(-3);b(3);
(%o26) 5
(%o27) false
```

2. Полная форма условного оператора:

```
(%i28) b1(a):=if a<0 then 5 else 10;
(%o28) b1(a):=if a<0 then 5 else 10
(%i29) b1(-3);b1(3);
(%o29) 5
(%o30) 10
```

3. Полная форма условного оператора с ветвлением:

```
(%i31) b2(a):=if a=9 then b:5 elseif a=10 then b:1 else b:0;
(%o31) b2(a):=if a=9 then b:5 elseif a=10 then b:1 else b:0
(%i32) b2(9);b2(10);b2(7);
(%o32) 5
(%o33) 1
(%o34) 0
```

4. Полный условный оператор, составленный с использованием логического оператора:

```
(%i35) y(x):=if (x<0 or x>=5) then x^2 else x^(1/2);  
(%o35) y(x):=if x<0\vx>=5 then x^2 else x^(1/2)  
(%i36) y(-3);y(7);y(3);  
(%o36) 9  
(%o37) 49  
(%o38)  $\sqrt{3}$ 
```

Оператор цикла может задаваться несколькими способами. Способ задания зависит от того, известно ли заранее, сколько раз необходимо выполнить тело цикла. Имеются следующие виды циклов:

for variable: initial_value step increment thru limit do body
for variable: initial_value step increment while condition do body
for variable: initial_value step increment unless condition do body

Как видно из их записи, операторы отличаются лишь последней частью, а их начало одинаково:

- *variable* – означает имя переменной, которая будет меняться внутри цикла;
- *initial_value* – начальное значение такой переменной;
- *increment* – то, на сколько изменяется эта переменная на каждой итерации цикла.

В циклах первого вида «*thru limit*» означает, что цикл должен закончиться тогда, когда переменная *variable* достигнет значения *limit*. В циклах второго вида «*while condition*» означает, что все операторы цикла будут выполняться до тех пор, пока верно условие *condition*. В случае «*unless condition*» цикл будет выполняться, пока условие *condition* ложно.

body – тело цикла.

При нормальном завершении цикла на экран выводится **done**. Принудительный выход из цикла осуществляется при помощи оператора **return**, который может возвращать произвольное значение. Для вывода результатов работы цикла существует команда **display()**.

Примеры:

В первом примере программа выводит все числа отрезка от 0 до 10, делящиеся на 2.

```
(%i14) for i:0 step 2 thru 10 do (b:i, display(b));
b=0
b=2
b=4
b=6
b=8
b=10
(%o14) done
```

Из этого примера видим, что несколько операторов можно объединять в блоки с помощью круглых скобок. Внутри скобок операторы следует разделять запятой.

Во втором примере приводится цикл для вычисления суммы натуральных чисел от 1 до 100.

```
(%i28) s:0$ for i:1 while i<=100 do s:s+i;
(%o29) done
(%i30) s;
(%o30) 5050
```

В третьем примере ищется список, составленный из максимальных элементов столбцов заданной матрицы. При решении данной задачи в программе нужно описать два массива:

- заданный двумерный массив (матрицу),
- одномерный массив, в который будут записываться максимальные элементы из каждого столбца. Пусть это матрица B размерности 5×5 , а искомый массив – это b .

```
(%i5) array(B,4,4); array(b,4);
(%o5) B
(%o6) b
```

Зададим элементы матрицы, например, следующим образом:

```
(%i7) B[i,j]:=i^2-j^3;
(%o7) Bi,j:=i2-j3
```

Для нахождения максимального элемента в столбцах потребуется второй внутренний (вложенный) цикл. Внешний цикл с переменной j будет перебирать столбцы, а во внутреннем цикле с переменной i элементы столбца будут сравниваться между собой для поиска наибольшего.


```

(%i14) for j:0 thru 4 do (b[j]:B[1,j], for i:1 thru 4 do
      if b[j]<B[i,j] then b[j]:B[i,j]);
(%o14) done
(%i15) display(b[0],b[1],b[2],b[3],b[4]);
b_0=16
b_1=15
b_2=8
b_3=-11
b_4=-48
(%o15) done

```

Как в условных операторах, так и в циклах вместо простых операторов можно писать составные операторы, т. е. блоки. Для этого имеется функция **block**($[v_1, \dots, v_m], \text{expr}_1, \dots, \text{expr}_n$). Здесь в квадратных скобках указаны локальные переменные, т. е. те, которые не будут доступны вне блока, а $\text{expr}_1, \dots, \text{expr}_n$ – последовательность операторов, которые будут выполняться внутри блока. Список локальных переменных может быть пустым. В качестве результата работы данная конструкция возвращает значение последнего выполненного оператора.

Пример:

Рассмотрим программу для вычисления знакопеременной суммы $S = \sum_{i=1}^n (-1)^i i$.

```

(%i1) SC(n):=block([sum,sig],sum:0,sig:-1,
      for i:1 thru n do (sum:sum+i*sig,sig:sig*(-1)),sum);
(%o1) SC(n):=block([sum,sig],sum:0,sig:-1,for i thru n
do (sum:sum+i sig,sig:sig(-1)),sum)

```

В данном случае пользователю не обязательно знать значение переменной sig , так как оно нужно лишь во время вычислений и не влияет на восприятие результата, поэтому данную переменную имеет смысл сделать внутренней.

В частности, имеем:

```

(%i12) SC(49);SC(50);SC(99);SC(100);
(%o12) -25
(%o13) 25
(%o14) -50
(%o15) 50

```

Функции для вычисления комбинаторных чисел. Для вычисления числа сочетаний и числа размещений в Maxima предусмотрены функции:

– **combination**(n, k) – находит число C_n^k сочетаний из n элементов по k ,

– **permutation**(n, k) – находит число A_n^k размещений из n элементов по k .

(Обе функции подключаются с помощью команды **load(functions)**.) Для вычисления числа сочетаний можно также использовать функцию **binomial**(x, y), которая определена на множестве всех действительных чисел. (Если x, y не являются целыми, то биномиальный коэффициент вычисляется как многочлен от x и y .)

Пример:

```
(%i1) load(functions)$  
      combination(8,5);permutation(8,5);  
(%o2) 56  
(%o3) 6720
```

Число P_n перестановок из n элементов вычисляет функция **factorial**(n). Удобно, однако, пользоваться функцией $n!$. Эта функция задает факториал в наиболее общем виде как гамма-функцию $x! = \text{gamma}(x+1)$ и определена для всех комплексных чисел, кроме целых отрицательных. Факториал от целых неотрицательных чисел автоматически упрощается ею до натурального числа.

Пример:

```
(%i4) factorial(10);  
(%o4) 3628800  
  
(%i5) 10!;  
(%o5) 3628800
```

Функции **stirling1**(n, k) и **stirling2**(n, k) позволяют находить числа Стирлинга первого и второго рода соответственно.

Пример:

```
(%i6) stirling1(8, 5);stirling2(8, 5);  
(%o6) -1960  
(%o7) 1050
```

Кроме рассмотренных функций в системе Maxima имеются другие функции для вычисления комбинаторных чисел. В частности, функции:

- **fib**(n), которая вычисляет n -е число Фибоначчи;
- **bern**(n), которая выполняет вычисление n -го числа Бернулли для целого n ;
- **euler**(n), которая выполняет вычисление n -го числа Эйлера для целого n .

Для решения рекуррентных соотношений используется пакет расширений **solve_rec**, который загружается с помощью команды **load(solve_rec)**.

Пример:

```
(%i10) load(solve_rec)$
       solve_rec(a[n+2]=2*a[n+1]+3*a[n], a[n]);

(%o11)  $a_n = k_1 3^n + k_2 (-1)^n$ 
```

Найдено общее решение линейного однородного рекуррентного соотношения второго порядка $a_{n+2} = 2a_{n+1} + 3a_n$.

Другие функции, которые потребуются при выполнении лабораторных работ. В ряде заданий лабораторной работы № 1 потребуется выполнять суммирование комбинаторных чисел данного вида. Для выполнения этой операции может быть использована команда **sum** ($expr, i, i_0, i_k$), которая суммирует заданное выражение в заданных пределах.

Примеры:

```
(%i2) sum(stirling2(8, k), k, 0, 8);
(%o2) 4140

(%i5) load(funcs)$
       sum(combination(8, k), k, 0, 8);

(%o6) 256
```

В лабораторной работе № 2 при нахождении частных решений линейных рекуррентных соотношений приходится решать системы линейных уравнений. Для этого в Maxima имеется команда **linsolve** ($[expr_1, \dots, expr_m], [x_1, \dots, x_n]$).

В wxMaxima – это команда **Solve Linear System** в меню **Уравнения** панели инструментов. После вызова команды на экране появляется окно, в котором указывается число уравнений системы

(рис. 1), а после указания числа уравнений – рабочее окно (рис. 2), в которое вводятся уравнения системы и переменные, относительно которых ее требуется решить.

Рис. 1

Рис. 2

Пример:

Решить систему:

$$\begin{cases} 2x + y + 3z = -9, \\ 8x + 3y + 5z = -13, \\ 2x + 5y - z = -5. \end{cases}$$

Вводя сначала число уравнений системы, а затем сами уравнения и переменные, относительно которых нужно решить систему, получим:

```
(%i10) linsolve([2*x+y+3*z=-9, 8*x+3*y+5*z=-13, 2*x+5*y-z=-5], [x,y,z]);
(%o10) [x=1, y=-2, z=-3]
```

В тех случаях, когда характеристическое уравнение линейного однородного рекуррентного соотношения с постоянными коэффициентами имеет комплексные корни, возможно, придется пользоваться командами **realpart()** и **imagpart()**, которые соответственно выделяют действительную и мнимую части комплексного числа. Возможно также, что для упрощения найденного общего или частного решения рекуррентного соотношения потребуются команды меню **Упростить** панели инструментов.

При решении задач методом производящих функций в лабораторной работе № 3 в подавляющем большинстве случаев будут востребованы всего две команды:

- **expand()**, которая раскрывает скобки и, следовательно, позволяет перемножать многочлены;

- **taylor(expr, x, a, n)**, которая выполняет разложение функции в ряд Тейлора в точке a (число n указывает степень члена ряда, на котором следует оборвать запись разложения).

Примеры:

```
(%i11) expand((x-3)*(x^2-3*x+4)*(x^3-4));
```

```
(%o11) x^6-6 x^5+13 x^4-16 x^3+24 x^2-52 x+48
```

```
(%i13) taylor(x^2/(1-x-x^2), x, 0, 8);
```

```
(%o13)/T/ x^2+x^3+2 x^4+3 x^5+5 x^6+8 x^7+13 x^8+...
```

ЛАБОРАТОРНАЯ РАБОТА № 1

Вычисление комбинаторных чисел, заданных формулами

В лабораторной работе № 1 формируются умения и навыки использования вычислительных возможностей системы Maxima при решении задач перечислительной комбинаторики на выборки элементов из заданного конечного множества [2, § 1.4, 1.5], на разбиения конечных множеств [3, § 1.12] и на перечисление подстановок [2, § 1.10]. Основное внимание уделено вычислению комбинаторных чисел, которые могут быть заданы в замкнутой форме.

1.1. Теоретическая справка

Выборки. Выборку k элементов из n элементов данного множества $A = \{a_1, \dots, a_n\}$ или кратко – (n, k) -выборку – можно однозначно задать списком элементов, включенных в эту выборку. Различают упорядоченные и неупорядоченные выборки. *Упорядоченной* называется выборка, в которой учитывается порядок размещения элементов. Если очередность следования элементов в выборке является несущественной, она называется *неупорядоченной*. Кроме того, выборки делят на *выборки без повторений* и *выборки с повторениями*.

Неупорядоченная (n, k) -выборка из множества A , все элементы в которой различны, называется *сочетанием из n элементов по k* , а неупорядоченная (n, k) -выборка, элементы в которой могут повторяться, – *сочетанием из n элементов по k с повторениями*.

Сочетание из n элементов по k – это k -элементное подмножество $\{a_{i_1}, \dots, a_{i_k}\}$ множества A , а (n, k) -сочетание с повторениями – k -элементное мультимножество, для которого множество A является основанием. При решении некоторых задач сочетания с повторениями удобно рассматривать и как k -элементные подмультимножества мультимножества, содержащего элементы n типов; в этом случае число элементов каждого типа должно быть не меньше k .

Число сочетаний из n элементов по k равно

$$C_n^k = \frac{n!}{k!(n-k)!} \quad (1.1)$$

при $k \leq n$ и $C_n^k = 0$ при $k > n$. Число сочетаний из n элементов по k с повторениями равно

$$\bar{C}_n^k = C_{n+k-1}^k. \quad (1.2)$$

Упорядоченная (n, k) -выборка из элементов множества A , все элементы которой различны, называется *размещением из n элементов по k* , а упорядоченная (n, k) -выборка, элементы которой могут повторяться, – *размещением k элементов из n с повторениями*.

Оба вида размещений являются элементами множества A^k , т. е. картежами $(a_{i_1}, \dots, a_{i_k})$ длины k , составленными из элементов множества A . В случае размещений без повторений все элементы

в картеже различны, а в случае размещений с повторениями могут повторяться.

Число размещений из n элементов по k равно

$$A_n^k = n(n-1) \dots (n-k+1) = \frac{n!}{(n-k)!} \quad (1.3)$$

при $k \leq n$ и $A_n^k = 0$ при $k > n$. Число размещений из n элементов по k с повторениями равно

$$\bar{A}_n^k = n^k. \quad (1.4)$$

Размещения из n элементов по n называются *перестановками из n элементов*. Перестановками из n элементов с повторениями называются картежи, содержащие все n элементов мультимножества

$$A = \{n_1 a_1, n_2 a_2, \dots, n_k a_k\}, n = \sum_{i=1}^k n_i.$$

Число перестановок из n элементов вычисляется по формуле

$$P_n = n(n-1) \dots 2 \cdot 1 = n!, \quad (1.5)$$

а число перестановок элементов мультимножества мощности n – по формуле

$$P_n(n_1, n_2, \dots, n_k) = \frac{n!}{n_1! n_2! \dots n_k!}. \quad (1.6)$$

Разбиения множеств. Пусть A – непустое конечное множество. Подмножества $A_i \subseteq A$ образуют разбиение множества A , если $A = A_1 \cup A_2 \cup \dots \cup A_k$ и $A_i \cap A_j = \emptyset$ при $i \neq j$. Подмножества A_i называются *блоками* разбиения, а их число k – *рангом* разбиения. Список $[n_1, n_2, \dots, n_k]$ мощностей $|A_i| = n_i$ блоков A_i называется *спецификацией* или *типом* разбиения, число n_i – *объемом* блока A_i .

Различают *упорядоченные* и *неупорядоченные* разбиения. Для упорядоченных разбиений порядок подмножеств в разбиении считается существенным, а для неупорядоченных – нет. Разбиение конечного множества A можно однозначно задать списком его блоков $A_i, i = 1, \dots, k$ и списками элементов, входящих в блоки.

Упорядоченное разбиение множества A на блоки A_i ($i = 1, \dots, k$) условимся обозначать, используя круглые скобки: (A_1, A_2, \dots, A_k) , а неупорядоченное – фигурные: $\{A_1, A_2, \dots, A_k\}$.

Наряду с разбиениями множеств на блоки в некоторых перечислительных задачах одновременно рассматриваются упорядочивания элементов в блоках разбиения.

При перечислении разбиений обычно прибегают к комбинаторной схеме, которую называют «раскладкой предметов по ящикам». Любое упорядоченное разбиение n -множества A ранга k интерпретируется как раскладка n различных предметов по k различным ящикам, а неупорядоченное разбиение – как раскладка n различных предметов по одинаковым (неразличимым) ящикам. Такие интерпретации особенно полезны при решении задач.

Основные вычислительные формулы для подсчета числа разбиений n -множества на k блоков приведены в таблице 1.

Таблица 1

<i>Тип раскладки n предметов по k ящикам</i>	<i>Размещение предметов в ящике</i>	<i>Число способов</i>
Ящики различные, в i -м ящике n_i предметов	Неупорядоченное	$P_n(n_1, n_2, \dots, n_k) = \frac{n!}{n_1! n_2! \dots n_k!}$
Ящики различные, допускаются пустые ящики	Неупорядоченное	$T(n, k) = k^n$
Ящики различные, нет пустых ящиков	Неупорядоченное	$T(n, k, 0) = \sum_{i=0}^k (-1)^i C_k^i (k-i)^n$
Ящики различные, имеется точно r пустых ящиков	Неупорядоченное	$T(n, k, r) = C_k^r \sum_{i=0}^{k-r} (-1)^i C_{k-r}^i (k-r-i)^n$
Ящики различные, допускаются пустые ящики	Упорядоченное	$n! C_{n+k-1}^n$
Ящики различные, нет пустых ящиков	Упорядоченное	$n! C_{n-1}^{k-1}$
Ящики неразличимые, в i -м ящике n_i предметов	Неупорядоченное	$\frac{1}{k!} P_n(n_1, n_2, \dots, n_k)$

Ящики неразличимые, нет пустых ящиков	Неупорядоченное	$S(n, k) = \frac{1}{k!} T(n, k, 0)$
Ящики неразличимые, допускаются пустые ящики	Неупорядоченное	$\sum_{i=0}^k S(n, i)$
Ящики неразличимые, нет пустых ящиков	Упорядоченное	$\frac{n!}{k!} C_{n-1}^{k-1}$

Комбинаторные числа

$$S(n, k) = \frac{1}{k!} \sum_{i=0}^k (-1)^i C_k^i (k-i)^n, \quad (1.7)$$

$S(n, 0) = 0$ при $n > 0$, $S(n, n) = 1$ при $n \geq 0$, $S(n, k) = 0$ при $k > n$, выражающие количество всех способов неупорядоченного разбиения n элементного множества на k непустых блоков, называются числами Стирлинга второго рода.

Числа $B(n)$, обозначающие число всех неупорядоченных разбиений n -множества на непустые блоки, называются числами Белла. Они связаны с числами Стирлинга второго рода соотношением

$$B(n) = \sum_{k=0}^n S(n, k), \quad (1.8)$$

при этом $B(0) = S(0, 0) = 1$.

Число всех неупорядоченных разбиений n -множества A на блоки $A_{i1}, A_{i2}, \dots, A_{in}$, $i = 1, 2, \dots, n$, среди которых для каждого i имеется $m_i \geq 0$ блоков с i элементами, т. е. $\sum_{i=1}^n i \cdot m_i = n$, равно

$$S_n(m_1, \dots, m_n) = \frac{n!}{m_1! \dots m_n! (1!)^{m_1} \dots (n!)^{m_n}}. \quad (1.9)$$

Перечисление подстановок. Подстановками называются биекции конечного множества на себя. Комбинаторные свойства подстановок любого n -множества удобно изучать на примере множества $I_n = \{1, \dots, n\}$. Подстановки множества I_n называют подстановками n -й степени, множество всех подстановками n -й степени обозначают S_n .

Подстановка $s \in S_n$ задается таблицей

$$\begin{pmatrix} 1 & 2 & \dots & n \\ i_1 & i_2 & \dots & i_n \end{pmatrix}, \quad (1.10)$$

в которой под числом $k \in I_n$ указывается его образ $i_k = s(k)$. Тождественное отображение множества I_n на себя, т. е. отображение, при котором $s(k) = k$ для каждого $k \in I_n$, определяется подстановкой $e = \begin{pmatrix} 1 & 2 & \dots & n \\ 1 & 2 & \dots & n \end{pmatrix}$; ее называют *тождественной подстановкой*. Для любой подстановки (1.10) существует *обратная* ей подстановка

$$s^{-1} = \begin{pmatrix} i_1 & i_2 & \dots & i_n \\ 1 & 2 & \dots & n \end{pmatrix} = \begin{pmatrix} 1 & 2 & \dots & n \\ s^{-1}(1) & s^{-1}(2) & \dots & s^{-1}(n) \end{pmatrix}.$$

На множестве S_n определено умножение подстановок. Произведением st подстановок s и t называется их последовательное выполнение, т. е. $st(i) = t(s(i))$ для всех $i \in I_n$.

Подстановка n -й степени называется *циклической* или *циклом*, если она элемент i_1 переводит в элемент i_2 , i_2 — в i_3 и т. д., i_{k-1} — в i_k , наконец, i_k — в i_1 , а остальные элементы оставляет на месте (т. е. переводит в себя). Такая подстановка обозначается (i_1, i_2, \dots, i_k) , при этом число $k \leq n$ называется *длиной цикла*. Два цикла n -й степени, не имеющие общих перемещаемых элементов, называются *независимыми*. Результат умножения независимых циклов не зависит от порядка сомножителей. Это позволяет любую подстановку $s \in S_n$ единственным образом представить в виде произведения попарно независимых циклов. При этом для элементов, которые переходят в себя, обычно вводят в рассмотрение циклы длины 1.

Если подстановка n -й степени s разлагается в произведение m_1 циклов длины 1, m_2 циклов длины 2 и т. д., m_n циклов длины n , то говорят, что она имеет *тип* $[1^{m_1}, 2^{m_2}, \dots, n^{m_n}]$. Число всех подстановок n -й степени типа $[1^{m_1}, 2^{m_2}, \dots, n^{m_n}]$ обозначим через $C(m_1, m_2, \dots, m_n)$, а число всех подстановок n -й степени, имеющих в разложении точно k циклов, — через $C(n, k)$. Имеют место формулы:

$$C(m_1, m_2, \dots, m_n) = \frac{n!}{1^{m_1} 2^{m_2} \dots n^{m_n} m_1! m_2! \dots m_n!}. \quad (1.11)$$

$$C(n, k) = \sum_{\substack{1 \cdot m_1 + 2 \cdot m_2 + \dots + n \cdot m_n = n, \\ m_1 + m_2 + \dots + m_n = k, \\ m_1 \geq 0, m_2 \geq 0, m_n \geq 0}} \frac{n!}{1^{m_1} 2^{m_2} \dots n^{m_n} m_1! m_2! \dots m_n!}. \quad (1.12)$$

Для решения широкого класса комбинаторных задач вместо чисел $C(n, k)$ удобнее рассматривать числа

$$s(n, k) = (-1)^{n+k} C(n, k), \quad (1.13)$$

которые называются *числами Стирлинга первого рода*. Числа $C(n, k)$ поэтому часто называют *числами Стирлинга первого рода без знака*. Очевидно, что $S(n, 0) = 0$ для $n > 0$, $s(n, n) = 1$ для всех $n \geq 0$ и $s(n, k) = 0$ для всех $k > n$.

Элемент $i \in I_n$ называется *неподвижным* (инвариантным) при подстановке s , если $s(i) = i$. Подстановка, не содержащая неподвижных элементов, называется *беспорядком*. Количество беспорядков среди подстановок n -й степени равно

$$D_n = n! \sum_{k=0}^n \frac{(-1)^k}{k!}. \quad (1.14)$$

Для $n = 0$ принимают $D_0 = 1$.

Подстановку, содержащую ровно k неподвижных элементов, называют *подстановкой с k встречами*. Число всех подстановок n -й степени с k встречами вычисляется по формуле

$$D(n, k) = C_n^k D_{n-k}. \quad (1.15)$$

1.2. Примеры решения задач с использованием Maxima

1. Рассмотрим примеры применения команд **combination**(n, k), **permutation**(n, k) и **factorial**(n) при решении задач на выборки.

Задача 1.1. Найти число десятизначных чисел, у которых все цифры различны.

Решение. Существует 10 цифр и 10! способов их перестановки. Среди этих перестановок будут перестановки, начинающиеся с цифры 0, таких перестановок – 9!. Поэтому искомый ответ: 10! – 9!. Проведем вычисления:

```
(%i1) factorial(10)-factorial(9);
(%o1) 3265920
```

Возможен и другой вариант выполнения вычислений:

```
(%i2) 10! - 9!;  
(%o2) 3265920
```

Ответ: 3265920.

Задача 1.2. Из 10 студентов и 8 студенток для участия в математической олимпиаде формируют команду, состоящую из 5 человек. Сколькими способами это можно сделать, если в команде должно быть не более 3 юношей?

Решение. В команде может не быть юношей, может быть 1, 2 или 3. Поэтому по правилу произведения и правилу суммы команду можно составить $C_{10}^0 C_8^5 + C_{10}^1 C_8^4 + C_{10}^2 C_8^3 + C_{10}^3 C_8^2$ способами. Вычисления:

```
(%i3) load(functions)$  
      combination(10,0)*combination(8,5)+combination(10,1)*combination(8,4)+  
      combination(10,2)*combination(8,3)+combination(10,3)*combination(8,2);  
(%o4) 6636
```

Ответ: 6636.

Задача 1.3. Найти число восьмизначных чисел с различными цифрами, кратных 5.

Решение. Число кратно 5, если оно оканчивается цифрой 0 или 5; расстановка остальных 7 цифр числа может быть произвольной. Поэтому имеется A_9^7 восьмизначных чисел с различными цифрами, у которых последняя цифра 0, и $A_9^7 - A_8^6$ чисел, у которых последняя цифра 5. По правилу суммы число всех восьмизначных чисел с различными цифрами, кратных 5, равно $2A_9^7 - A_8^6$. Вычисления:

```
(%i5) load(functions)$  
      2*permutation(9,7)-permutation(8,6);  
(%o6) 342720
```

Ответ: 342720.

Задача 1.4. Найти число восьмизначных чисел, у которых три цифры нечетные, а остальные четные.

Решение. Имеется 5 четных и 5 нечетных цифр. Поэтому любую цифру числа, кроме первой, можно выбрать пятью способами. Позиции для четных цифр можно выбрать C_8^5 способами, в C_7^4 случаях из них на первом месте будет стоять нуль. Значит, искомое число равно $(C_8^5 - C_7^4) \cdot 5^5 \cdot 5^3 = (C_8^5 - C_7^4) \cdot 5^8$. Вычисления:

```
(%i3) load(funcs)$
      (combination(8,5)- combination(7,4))*5^8;
(%o4) 8203125
```

Ответ: 8203125.

Задача 1.5. В продаже имеются маркеры четырех цветов. Сколькими способами можно купить 15 маркеров так, чтобы в покупке было хотя бы по одному маркеру каждого цвета?

Решение. Если сразу взять по одному маркеру каждого цвета, то из имеющегося ассортимента останется выбрать еще 11 маркеров. Этот выбор может быть сделан $\bar{C}_4^{11} = C_{4+11-1}^{11} = C_{14}^{11}$ способами. Вычисления:

```
(%i5) load(funcs)$
      combination(14,11);
(%o6) 364
```

Ответ: 364.

Задача 1.6. Сколько различных десятизначных чисел, кратных 9, можно составить из чисел 1, 2 и 3, если в каждом из них цифра 3 должна повторяться дважды?

Решение. Число кратно 9, если сумма его цифр делится на 9. Среди 10 цифр числа, удовлетворяющего условиям задачи, две цифры 3. Пусть среди остальных восьми цифр числа имеется k единиц, а значит, $(8 - k)$ двоек ($k \leq 8$). Тогда на 9 делится число $k + 2(8 - k) + 2 \cdot 3 = 22 - k$. Это возможно лишь при $k = 4$. Поэтому в каждое число, которое удовлетворяет заданным условиям, цифры 1 и 2 входят по четыре раза. Следовательно, иско-
мых чисел $P_{10}(4,4,2) = \frac{10!}{4!4!2!}$. Вычисления:

```
(%i8) 10!/(4!*4!*2!);
(%o8) 3150
```

Ответ: 3150.

2. Дадим примеры применения команда **stirling2**(n, k) для вычисления чисел Стирлинга второго рода $S(n, k)$ в задачах на разбиения множеств. Из таблицы 1 видим, что эту команду можно также применять для перечисления числа:

- упорядоченных разбиений n -множества на k блоков, среди которых нет пустых: $T(n, k, 0) = k! S(n, k)$;
- упорядоченных разбиений n -множества на k блоков, среди которых r пустых: $T(n, k, r) = \frac{k!}{r!} S(n, k - r)$.

Для вычисления числа $\sum_{i=0}^k S(n, i)$ всех неупорядоченных разбиений n -множества на k блоков, среди которых допускаются пустые, и чисел Белла $B(n)$ может быть использована команда **sum** (*expr*, *i*, *i_0*, *i_1*), которая суммирует заданное выражение в заданных пределах. Как проводить вычисления в остальных случаях, мы уже рассмотрели в предыдущем пункте.

Задача 1.7. Сколькими способами 12 различных открыток можно разложить в 8 одинаковых конвертов, если

- а) в каждом конверте должна лежать хотя бы одна открытка;
- б) допускаются пустые конверты?

Решение. а) Требуется найти число неупорядоченных разбиений множества из 12 элементов на 8 непустых блоков, т. е. найти $S(12, 8)$. Вычисления:

```
(%i1) stirling2(12, 8);
(%o1) 159027
```

- б) Теперь нужно вычислить сумму $\sum_{i=0}^8 S(12, i)$. Имеем:

```
(%i2) sum(stirling2(12, i), i, 0, 8);
(%o2) 4189550
```

Ответ: а) 159027, б) 4189550.

Задача 1.8. Найти число всех способов раскладки 12 различных открыток в одинаковые конверты (пустые конверты не допускаются).

Решение. Задача сводится к вычислению числа Белла $B(12)$. Вычисления:

```
(%i4) sum(stirling2(12, i), i, 0, 12);
(%o4) 4213597
```

Ответ: 4213597.

Задача 1.9. Сколькими способами 12 различных открыток можно разложить в 8 различных конвертов, если

- а) в каждом конверте должна лежать хотя бы одна открытка;
- б) допускаются пустые конверты;
- в) должно остаться два пустых конверта?

Решение. а) Требуется найти число $T(12, 8, 0)$ упорядоченных разбиений множества из 12 элементов на 8 непустых блоков, но $T(12, 8, 0) = 8! S(12, 8)$. Имеем:

```
(%i5) 8!*stirling2(12, 8);
(%o5) 6411968640
```

либо

```
(%i7) 8!*(%o1);
(%o7) 6411968640
```

б) Требуется найти число $T(12,8)$ упорядоченных разбиений множества из 12 элементов на 8 блоков, среди которых могут быть пустые, это число 8^{12} . Вычисляем:

```
(%i8) 8^12;
(%o8) 68719476736
```

в) Ищется число упорядоченных разбиений множества из 12 элементов на 8 блоков, среди которых ровно 2 пустых, т. е. число $T(12,8,2) = \frac{8!}{2!} S(12,8-2) = \frac{8!}{2!} S(12,6)$. Имеем:

```
(%i9) (8!/2!)*stirling2(12, 6);
(%o9) 26684824320
```

Ответ: а) 6411968640, б) 68719476736, в) 26684824320.

Задача 1.10. Сколькими способами 12 различных открыток можно разложить в одинаковые конверты так, чтобы в двух конвертах лежало по одной открытке, в трех – по две открытки и одном – четыре?

Решение. Чтобы ответить на поставленный вопрос, нужно найти число неупорядоченных разбиений множества из 12 элементов, которые имеют тип $[1^2, 2^0, 3^2, 4^1, 5^0, 6^0, \dots, 12^0]$. Это число равно

$\frac{12!}{2!2! \cdot 1!(1!)^2(3!)^2(4!)^1}$. Вычисления дают:

```
(%i24) 12!/((2!)^2*(3!)^2*4!);
(%o24) 138600
```

Ответ: 138600.

3. Команда **stirling1**(n, k) для вычисления чисел Стирлинга первого рода $s(n, k)$ позволяет находить число $C(n, k)$ всех подстановок n -й степени, имеющих в своем разложении точно k циклов. Нахождение числа $C(m_1, m_2, \dots, m_n)$ всех подстановок n -й степени данного типа $[1^{m_1}, 2^{m_2}, \dots, n^{m_n}]$ сводится к арифметическим вычислениям. Для вычисления числа беспорядков D_n можно использовать команду **sum**.

Рассмотрим примеры.

Задача 1.11. Найти число всех подстановок 12-й степени, в разложении которых

а) два одночленных цикла, два цикла длины 3 и один цикл длины 4;

б) ровно 5 циклов.

Решение. а) Требуется найти число подстановок 12-й степени, которые имеют тип $[1^2, 2^0, 3^2, 4^1, 5^0, 6^0, \dots, 12^0]$. Это число равно

$\frac{12!}{1^2 3^2 4^1 2! 2! 1!}$. Вычисления дают:

```
(%i28) 12!/(3^2*4*(2!)^2);
```

```
(%o28) 3326400
```

б) Решение сводится к вычислению числа $C(12,5) = |s(12,5)|$. Вычисления дают:

```
(%i30) abs(stirling1(12, 5));
```

```
(%o30) 45995730
```

Ответ: а) 3326400, б) 45995730.

Задача 1.12. Девять джентльменов при входе в ресторан сдают свои шляпы в гардероб. Сколькими способами рассеянный гардеробщик может выдать при выходе шляпы так, чтобы

а) ни один из джентльменов не получил свою шляпу;

б) только три джентльмена получили свои шляпы;

в) хотя бы один из джентльменов получил свою шляпу?

Решение. а) Задача сводится к вычислению числа беспорядков

$D_9 = 9! \sum_{k=0}^9 \frac{(-1)^k}{k!}$. Имеем:

```
(%i10) 9!*sum((-1)^k/k!, k, 0, 9);
```

```
(%o10) 133496
```

б) Необходимо найти число $D(9,3) = C_9^3 D_{9-3} = C_9^3 D_6$. Вычисления дают:

```
(%i18) load(funcs)$  
combination(9, 3)*6!*sum((-1)^k/k!, k, 0, 6);
```

```
(%o19) 22260.0
```

в) Число всех способов, которыми гардеробщик может выдать шляпы, равно $8!$, при этом только в D_9 случаях ни один из джентльменов не получит свою шляпу. Поэтому искомое число равно разности $8! - D_9$. Имеем:


```
(%i22) 9!-(%o10);  
(%o22) 229384
```

Ответ: а) 133496, б) 22260, в) 229384.

1.3. Задачи для самостоятельного решения

Решить следующие задачи. При выполнении вычислений использовать систему Махима.

1. Сколько среди перестановок цифр числа 123456789 таких, которые начинаются: а) с двух четных цифр, б) с двух цифр, меньших 6?

2. Сколько среди четырехзначных чисел таких, в записи которых: а) все цифры различные, б) все цифры различные и хотя бы одна цифра четная, в) все цифры различные и ровно одна цифра четная, г) хотя бы одна цифра четная?

3. Сколько семизначных чисел можно составить из цифр 0, 1, 2, 3, 4, 5 и 6, в записи которых: а) все цифры различные, б) нет цифры 0, в) дважды повторяется цифра 0, а остальные цифры различные?

4. Сколько различных анаграмм можно составить из букв слова «обобщение»?

5. Имеется колода из 52 карт. Сколькими способами можно вытянуть 10 карт, среди которых: а) 4 туза; б) хотя бы один туз; в) два туза, одна дама, одна бубновая карта; г) ровно пять карт одной масти?

6. Рота состоит из 3 офицеров, 6 сержантов и 60 рядовых. Сколькими способами можно составить наряд, состоящий из одного офицера, двух сержантов и 20 рядовых?

7. В аквариуме плавают 12 гуппи, 10 меченосцев и 6 гурами. Надо пересадить в другой аквариум не менее 4, но не более 6 гуппи, не более 4 меченосцев и 2 гурами. Сколькими способами это можно сделать?

8. В цветочном магазине продается 8 видов цветов. Сколькими способами можно составить букет: а) из 15 цветов; б) из 15 цветов, если в нем должны быть представлены все виды цветов?

9. Сколькими способами из десяти супружеских пар можно выбрать комиссию из четырех человек так, чтобы она не содержала ни одной супружеской пары.

10. В почтовом киоске имеются открытки 12 сортов. Сколькими способами можно купить: а) 12 открыток, б) 8 открыток, в) 8 разных открыток?

11. Сколькими способами можно расставить 20 различных книг в книжном шкафу с шестью полками, если каждая полка может вместить все 20 книг?

12. Решить предыдущую задачу при условии, что на каждую полку должна быть поставлена хотя бы одна книга.

13. Переплетчик должен переплести 12 различных дипломных работ. Сколькими способами он может это сделать, если у него в наличии имеется бумага трех цветов?

14. Решить предыдущую задачу при условии, что в бумагу каждого цвета должен быть переплетен хотя бы один диплом.

15. Переплетчик, у которого в наличии имеется бумага красного, синего и зеленого цветов, должен переплести 12 различных дипломных работ. Сколькими способами он может это сделать, если два диплома собирается переплести в красную бумагу, три – в синюю, а остальные – в зеленую?

16. Компания, состоящая из 12 супружеских пар, разбивается на 6 групп по 4 человека для лодочной прогулки. Сколькими способами можно разбить их так, чтобы в каждой лодке оказалось два мужчины и две женщины?

17. Сколькими способами можно разместить 14 различных предметов в 4 ящиках, если: а) ящики различимы и могут быть пустыми, б) ящики различимы и не могут быть пустыми, в) ящики неразличимы и могут быть пустыми, г) ящики неразличимы и не могут быть пустыми?

18. Сколькими способами студенческую группу из 24 человек можно разбить на 4 команды для игры в волейбол?

19. Сколькими способами можно разложить 9 различных книг в три бандероли по три книги в каждой?

20. У преподавателя обучается 25 студентов. Преподаватель проводит тест, а затем предлагает студентам обменяться работами для взаимопроверки. Сколькими способами это можно сделать?

21. Неопытный официант подает 14 блюд, обслуживая 14 клиентов. Сколькими способами он может сделать так, чтобы: а) ни один клиент не получил свой заказ, б) только половина клиентов получила свои заказы?

22. Матрицей перестановок размерности n называется матрица, в каждой строке и в каждом столбце которой только один элемент равен 1, а все остальные равны 0. Сколько существует матриц перестановок размерности 8×8 , содержащих на главной диагонали только нули?

23. Сколькими способами можно переставить цифры в числе 123456789 так, чтобы: а) ни одна цифра не стояла на своем месте, б) ни одна четная цифра не стояла на своем месте, в) ровно три цифры остались на своих местах?

24. Найти число всех подстановок 10-й степени, в разложении которых: а) три одночленных цикла, два цикла длины 2 и один цикл длины 3; б) ровно 3 циклов.

1.4. Варианты индивидуальных заданий

Номер варианта	Задачи
1.	1, 5, 11, 17, 20, 24
2.	2, 6, 12, 17, 21, 24
3.	3, 7, 13, 17, 22, 24
4.	4, 8, 14, 17, 23, 24
5.	5, 9, 15, 17, 20, 24
6.	6, 10, 16, 17, 21, 24
7.	7, 1, 18, 17, 22, 24
8.	8, 2, 19, 17, 23, 24
9.	9, 3, 13, 17, 20, 24
10.	10, 4, 14, 17, 21, 24

ЛАБОРАТОРНАЯ РАБОТА № 2

Рекуррентные соотношения

В лабораторной работе № 2 формируются умения и навыки рекурсивных вычислений [2, § 2.1] и решения линейных рекуррентных соотношений [2, § 2.2, 2.3].

2.1. Теоретическая справка

Рекурсивные вычисления. Часто при подсчете комбинаторных чисел гораздо удобнее прибегать к рекурсии, чем пользоваться явными формулами для их вычисления. Применяя этот способ, находят одно или несколько первых комбинаторных чисел требуемого вида и указывают правило (рекуррентное соотношение), позволяющее вычислить по ним последующие числа.

Рекуррентное соотношение

$$u_{n+k} = \Phi(n; u_n, \dots, u_{n+k-1}), \quad (2.1)$$

где $n, k \in \mathbf{N}_0$, называется *рекуррентным соотношением* или *рекуррентным уравнением порядка k* . Если найдены (заданы) k первых комбинаторных чисел u_0, u_1, \dots, u_{k-1} , то, используя соотношение (2.1), можно вычислить следующее комбинаторное число u_k и вообще число u_n для любого $n \in \mathbf{N}_0$.

Приведем несколько примеров известных рекуррентных соотношений.

1. Рекуррентное соотношение первого порядка

$$P_{n+1} = (n+1)P_n, P_0 = 1 \quad (2.2)$$

для числа P_n перестановок из n элементов.

2. Рекуррентное соотношение первого порядка

$$D_{n+1} = (n+1)D_n + (-1)^{n+1}, D_0 = 1 \quad (2.3)$$

для числа беспорядков D_n среди подстановок n -й степени.

3. Линейное однородное рекуррентное соотношение второго порядка

$$F_{n+2} = F_{n+1} + F_n, F_0 = 1, F_1 = 1 \quad (2.4)$$

для чисел Фибоначчи.

4. Рекуррентное соотношение

$$C_n^k = C_{n-1}^{k-1} + C_{n-1}^k, C_n^0 = C_n^n = 1 \text{ при } n \geq 0, C_n^k = 0 \text{ при } k > n$$

для числа сочетаний из n элементов по k .

5. Рекуррентное соотношение

$s(n, k) = s(n - 1, k - 1) - (n - 1)s(n - 1, k)$,
 $s(n, n) = 1$ при $n \geq 0$, $s(n, 0) = 0$ при $n > 0$ и $s(n, k) = 0$ при $k > n$ для чисел Стирлинга первого рода.

6. Рекуррентное соотношение

$S(n, k) = S(n - 1, k - 1) + kS(n - 1, k)$
 $S(n, n) = 1$ при $n \geq 0$, $S(n, 0) = 0$ при $n > 0$ и $S(n, k) = 0$ при $k > n$ для чисел Стирлинга второго рода.

7. Нелинейное рекуррентное соотношение

$c_n = c_0 c_{n-1} + c_1 c_{n-2} + \dots + c_{n-1} c_0$, $c_0 = 1$
 для чисел Каталана. В вычислительной практике гораздо чаще используется более простое линейное однородное рекуррентное соотношение первого порядка

$$c_{n+1} = \frac{4n+2}{n+2} c_n, c_0 = 1.$$

Решение рекуррентных соотношений. В некоторых случаях, наоборот, более целесообразно от рекуррентного задания комбинаторных чисел переходить к их заданию с помощью явных формул. Для этого рекуррентное соотношение рассматривают как уравнение и решают его при заданных начальных условиях.

Функция $u_n = f(n)$ называется *решением* рекуррентного соотношения (2.1), если при ее подстановке в это соотношение оно для каждого $n \in \mathbf{N}_0$ обращается в верное числовое равенство.

Соотношение (2.1) связывает $k + 1$ значений функции $f(n)$. Если известны k ее первых значений (*начальные условия*)

$$u_0 = f(0), u_1 = f(1), \dots, u_{k-1} = f(k-1), \quad (2.5)$$

где $f(0), f(1), \dots, f(k-1)$ – заданные числа, то можно найти $f(k)$ и вообще $f(n)$ для любого n . Поэтому решение $f(n)$ можно также записать в виде

$$f(n) = F(n; f(0), f(1), \dots, f(k-1)). \quad (2.6)$$

Другим начальным значениям будет соответствовать другое решение соотношения (2.1), такие решения называются его *частными* решениями.

Частное решение (2.6) задает числовую последовательность

$$u_0 = f(0), u_1 = f(1), \dots, u_{k-1} = f(k-1), \dots, u_n = f(n), \dots$$

Эту последовательность называют *рекуррентной последовательностью порядка k* , заданной соотношением (2.1), и также считают решением этого рекуррентного соотношения.

Частные решения объединяет функция

$$f(n) = F(n; C_0, C_1, \dots, C_{k-1}), \quad (2.7)$$

где C_0, C_1, \dots, C_{k-1} — произвольные числа. Решение (2.7) называется *общим* решением соотношения (2.1).

Общих универсальных методов решения рекуррентных соотношений не существует. Серьезные трудности могут возникнуть уже при решении линейных рекуррентных соотношений первого порядка.

Линейным рекуррентным соотношением первого порядка называется соотношение вида

$$u_{n+1} = a_n u_n + b_n, \quad (2.8)$$

где a_n, b_n — заданные вещественные или комплексные функции аргумента $n \in N_0$; причем $a_n \neq 0$ для всех $n \in N_0$.

Если $b_n = 0$ для каждого $n \in N_0$, то линейное однородное соотношение (2.8) называется *однородным*; в противном случае оно называется *линейным неоднородным*.

Линейное однородное рекуррентное соотношение первого порядка обычно решается *методом последовательных подстановок*. Использование этого метода сопряжено с вычислениями конечных сумм и произведений, порой достаточно громоздкими и сложными. При решении линейных неоднородных рекуррентных соотношений первого порядка, кроме того, приходится использовать *метод вариации постоянных* [2, § 2.2].

Наиболее разработана теория решения линейных однородных рекуррентных соотношений с постоянными коэффициентами [2, § 2.3, 2.4].

Линейным однородным рекуррентным соотношением порядка k с постоянными коэффициентами называется соотношение

$$u_{n+k} = a_1 u_{n+k-1} + a_2 u_{n+k-2} + \dots + a_k u_n, \quad n \in N_0, \quad (2.9)$$

где a_1, a_2, \dots, a_k — заданные числа (действительные или комплексные), причем $a_k \neq 0$.

Общее решение соотношения (2.9) определяется корнями его *характеристического уравнения*

$$\lambda^k = a_1 \lambda^{k-1} + \dots + a_{k-1} \lambda + a_k. \quad (2.10)$$

Если характеристическое уравнение (2.10) соотношения (2.9) имеет k различных корней $\lambda_1, \lambda_2, \dots, \lambda_k$, то общее решение этого соотношения имеет вид

$$f(n) = C_1 \lambda_1^n + C_2 \lambda_2^n + \dots + C_k \lambda_k^n.$$

В общем случае характеристическое уравнение (2.10) может иметь r различных корней: λ_1 кратности k_1 , λ_2 кратности k_2 , \dots , λ_r кратности k_r , $\sum_{i=1}^r k_i = k$. Тогда в общем решении соотношения (2.9) каждый корень λ_i представлен слагаемым

$$f_i(n) = (C_{i1} + C_{i2}n + \dots + C_{ik_i}n^{k_i-1})\lambda_i^n,$$

$i = 1, 2, \dots, r$, а само общее решение есть сумма таких слагаемых.

Нередко приходится сталкиваться с решением соотношений (2.9) с действительными коэффициентами, когда среди корней характеристического уравнения имеются комплексные, а решение соотношения требуется записать без использования степеней с комплексными основаниями. Если коэффициенты рекуррентного соотношения действительны, а характеристическое уравнение имеет комплексный корень $a + bi$ кратности l , то его l -кратным корнем является и сопряженный ему корень $a - bi$. Эти корни в общем решении представляются суммой

$$r^n(C_1 + nC_2 + \dots + n^{l-1}C_l)\cos n\varphi + r^n(C_{l+1} + nC_{l+2} + \dots + n^{l-1}C_{2l})\sin n\varphi,$$

где $a + bi = r(\cos\varphi + i\sin\varphi)$.

При решении линейных однородных рекуррентных соотношений «с бумагой и ручкой» основные трудности возникают при решении алгебраического уравнения (2.10). Из-за громоздких вычислений затруднения, кроме того, могут возникнуть в решении систем линейных уравнений при нахождении частных решений соотношения (2.9), соответствующих заданным начальным условиям.

Рекуррентное соотношение

$$u_{n+k} = a_1 u_{n+k-1} + a_2 u_{n+k-2} + \dots + a_k u_n + b_n, \quad (2.11)$$

где a_1, a_2, \dots, a_k – заданные числа (действительные или комплексные), причем $a_k \neq 0$, а b_n – заданная функция, называется *линейным неоднородным рекуррентным соотношением порядка k с постоянными коэффициентами*; а рекуррентное соотношение (2.9) – *однородным рекуррентным соотношением*, соответствующим неоднородному соотношению (2.11). Общее решение неоднородного линейного рекуррентного соотношения (2.11) является

суммой любого его частного решения $g(n)$ и общего решения соответствующего ему однородного соотношения (2.9).

Основные трудности при решении линейных неоднородных рекуррентных соотношений связаны с отысканием их частных решений; они, как правило, ищутся методом вариации постоянных. Наибольший интерес представляют случаи:

1. $b_n = \beta^n$.

Если β не является корнем характеристического уравнения (2.10), то частное решение соотношения (2.11) ищется в виде $g(n) = C\beta^n$, если же β – r -кратный корень уравнения (2.10), то в виде $g(n) = Cn^r\beta^n$.

2. $b_n = p(n)$, где $p(n) = p_0n^r + p_1n^{r-1} + \dots + p_r$ – многочлен от n степени r .

Если 1 не является корнем характеристического уравнения (2.10), частное решение соотношения (2.11) находится в виде $g(n) = C_0 + C_1n + \dots + C_rn^r$, если же 1 – корень уравнения (2.6), то в виде $g(n) = n^m \sum_{i=0}^r C_i n^i$, где m – кратность корня 1.

3. $b_n = \sin(\alpha n)$ либо $b_n = \cos(\alpha n)$.

В этом случае обычно пользуются тем, что если соотношение

$$u_{n+k} = a_1 u_{n+k-1} + a_2 u_{n+k-2} + \dots + a_k u_n + A_n + iB_n,$$

где a_1, a_2, \dots, a_k и функции A_n, B_n действительны, имеет решение $g(n) = a(n) + ib(n)$, то его действительная часть $a(n)$ и мнимая часть $b(n)$ являются соответственно решениями соотношений

$$u_{n+k} = a_1 u_{n+k-1} + a_2 u_{n+k-2} + \dots + a_k u_n + A_n,$$

$$u_{n+k} = a_1 u_{n+k-1} + a_2 u_{n+k-2} + \dots + a_k u_n + B_n.$$

4. Если $g(n), \bar{g}(n)$ – частные решения линейных неоднородных рекуррентных соотношений

$$u_{n+k} = a_1 u_{n+k-1} + \dots + a_k u_n + b_n,$$

$$u_{n+k} = a_1 u_{n+k-1} + \dots + a_k u_n + \bar{b}_n,$$

соответственно, то $C_1 g(n) \pm C_2 \bar{g}(n)$ – частное решение рекуррентного соотношения

$$u_{n+k} = a_1 u_{n+k-1} + \dots + a_k u_n + (C_1 b_n \pm C_2 \bar{b}_n).$$

2.2. Примеры решения задач с использованием Maxima

1. Дадим примеры программ на встроенном макроязыке системы Maxima для вычисления комбинаторных чисел, заданных рекуррентными соотношениями.

Задача 2.1. На встроенном макроязыке системы Maxima написать программу для вычисления числа P_n перестановок из n элементов. Найти: а) P_{10} , б) P_{15} .

Решение. Из соотношения (2.2) следует, что для вычисления значения P_n используется лишь предыдущее P_{n-1} и номер итерации n . Следовательно, чтобы найти значение P_n в функции потребуется написать цикл для всех $k = 1, \dots, n$, а внутри цикла результат вычислений можно записывать в одну и ту же переменную на каждой итерации, так как это значение не влияет на вычисления, кроме тех, которые будут произведены на следующей итерации. Поэтому, учитывая, что $P_0 = 1$, приходим к функции:

```
(%i16) P(n):=block ([p],p: 1,for i: 1 thru n do (p: i*p),p);  
(%o16) P(n):=block([p],p:1,for i thru n do p:i*p,p)
```

Для $n = 10$, $n = 15$ имеем:

```
(%i17) P(10);P(15);  
(%o17) 3628800  
(%o18) 130767436800
```

Ответ: а) $P_{10} = 3628800$, б) $P_{15} = 130767436800$.

Задача 2.2. На встроенном макроязыке системы Maxima написать программу для вычисления числа беспорядков D_n . Найти: а) D_{10} , б) D_{15} .

Решение. Рекуррентное соотношение (2.3) для числа беспорядков отличается от предыдущего (2.2) лишь нелинейным слагаемым $(-1)^{n+1}$. Поэтому в теле цикла нужно учесть это слагаемое. Для вычисления $(-1)^n$ можно использовать два способа: условный оператор if и умножение на -1 на каждом шаге цикла. Однако более удобно воспользоваться встроенной функцией $\text{signum}(x)$ для $x = (-1)^i$. При этом функция для вычисления числа беспорядков D_n будет иметь вид:

```
(%i19) D(n):=block ([d],d: 1,for i: 1 thru n
do (d: i*d+signum((-1)^i)),d);
(%o19) D(n):=block([d],d:1,for i thru n do d:i*d+signum((-1)^i),d)
```

Для $n = 10$, $n = 15$ получим:

```
(%i20) D(10);D(15);
(%o20) 1334961
(%o21) 48106651573
```

Ответ: а) $D_{10} = 1334961$, б) $D_{15} = 48106651573$.

Задача 2.3. На встроенном макроязыке системы Maxima написать программу для вычисления чисел Фибоначчи F_n . Найти: а) F_{10} , б) F_{15} .

Решение. Каждый член последовательности Фибоначчи F_n (2.4) зависит от двух предыдущих, поэтому при составлении программы потребуется задать два начальных условия $F_0 = 1$, $F_1 = 1$. Ориентируясь на предыдущие примеры, сделаем соответствующий «сдвиг» значений, позволяющий хранить в памяти два значения последовательности. Таким образом, легко получается следующая функция:

```
(%i22) F(n):=block ([ft,f0,f1],f0:1,f1:1,
for i: 2 thru n do (ft: f1+f0,f0: f1,f1: ft),ft);
(%o22) F(n):=block
([ft,f0,f1],f0:1,f1:1,for i from 2 thru n do (ft:f1+f0,f0:f1,f1:ft),f
```

Для $n = 10$, $n = 15$ имеем:

```
(%i23) F(10);F(15);
(%o23) 89
(%o24) 987
```

Ответ: а) $F_{10} = 89$, б) $F_{15} = 987$.

Примечание. Построенная функция позволяет вычислять n -й член последовательности Фибоначчи. Но если в этой функции менять начальные условия, то можно аналогичным образом вычислить члены других последовательностей, заданных рекуррентным соотношением (2.4). Можно также составить функцию, в которой в качестве параметров будут передаваться начальные значения последовательности:

```
(%i25) FP(f0,f1,n):=block([ft],
    for i: 2 thru n do (ft: f1+f0,f0:f1,f1: ft),ft);
(%o25) FP(f0,f1,n):=
    block([ft],for i from 2 thru n do (ft:f1+f0,f0:f1,f1:ft),f
```

Новая функция является более универсальной и позволяет вычислять члены последовательности (2.4) с другими начальными условиями. Например, для последовательности *чисел Люка*, которая определяется соотношением $L_{n+2} = L_{n+1} + L_n$, $L_0 = 1$, $L_1 = 3$, получим:

```
(%i26) FP(1,3,2);FP(1,3,3);FP(1,3,4);FP(1,3,5);
(%o26) 4
(%o27) 7
(%o28) 11
(%o29) 18
```

Следовательно, $L_2 = 4$, $L_3 = 7$, $L_4 = 11$, $L_5 = 18$.

Задача 2.4. Написать программу для вычисления биномиальных коэффициентов C_n^k . Найти: а) C_{15}^{10} , б) C_{20}^{10} .

Решение. Для программной реализации рекуррентного соотношения $C_n^k = C_{n-1}^{k-1} + C_{n-1}^k$, $C_n^0 = C_n^n = 1$ при $n \geq 0$ и $C_n^k = 0$ при $k > n$ будем использовать два цикла: внешний цикл по n и внутренний – по k .

Для $n = 15$, $k = 10$ и $n = 20$, $k = 10$ соответственно получим:

```
(%i16) NC(k,n) :=
    block ( [Arr,tmp], array(Arr,if k>0 then k
    else 1), Arr[0]:1, Arr[1]:1, for i: 2 thru k do Arr[i]:0,
    for i: 2 thru n do (for j: k step -1 thru 1 do
    Arr[j]: Arr[j]+Arr[j-1],tmp:Arr[k]+0);
(%o16) NC(k,n):=block([Arr,tmp],array(Arr,if k>0 then k else 1),Arr[0]:1,
Arr[1]:1,for i from 2 thru k do Arr[i]:0,for i from 2 thru n do for j from
k step -1 thru 1 do Arr[j]:Arr[j]+Arr[j-1],tmp:Arr[k]+0)
(%i17) NC(10,15);NC(10,20);
(%o17) 3003
(%o18) 184756
```

Ответ: а) $C_{15}^{10} = 3003$, б) $C_{20}^{10} = 184756$.

Задача 2.5. Написать программу для вычисления чисел $s(n, k)$ Стирлинга первого рода. Найти: а) $s(10,4)$, б) $s(15,7)$.

Решение. Числа Стирлинга первого рода определяются соотношением $s(n, k) = s(n-1, k-1) - (n-1)s(n-1, k)$, $s(n, n) = 1$ при $n \geq 0$, $s(n, 0) = 0$ при $n > 0$ и $s(n, k) = 0$ при $k > n$. Вновь требуется использовать два цикла: внешний цикл по n и внутренний – по k .

```
(%i40) NS(k,n):=
  block ([Arr,tmp], array(Arr,if k>0 then k
    else 1), Arr[0]:0, Arr[1]:1, for i: 2 thru k do Arr[i]:0,
    for i: 2 thru n do (for j: k step -1 thru 1 do
      Arr[j]: (i-1)*Arr[j]+Arr[j-1]),tmp:Arr[k]+0);
(%o40) NS(k,n):=block ([Arr,tmp], array(Arr,if k>0 then k else 1), Arr_0:0,
  Arr_1:1, for i from 2 thru k do Arr_i:0, for i from 2 thru n do for j fr
    k step -1 thru 1 do Arr_j:(i-1)Arr_j+Arr_{j-1}, tmp:Arr_k+0)
```

Отсюда

```
(%i41) NS(4, 10); NS(7, 15);
(%o41) 723680
(%o42) 14409322928
```

Ответ: а) $s(10, 4) = 723680$, б) $s(15, 7) = 14409322928$.

Задача 2.6. Написать программу для вычисления чисел $S(n, k)$ Стирлинга второго рода. Найти: а) $S(10, 4)$, б) $S(15, 7)$.

Решение. Числа Стирлинга 2-го рода определяются соотношением $S(n, k) = S(n-1, k-1) + kS(n-1, k)$, $S(n, n) = 1$ при $n \geq 0$, $S(n, 0) = 0$ при $n > 0$ и $S(n, k) = 0$ при $k > n$.

Здесь

```
(%i43) NS(k,n):=
  block ([Arr,tmp], array(Arr,if k>0 then k
    else 1), Arr[0]:0, Arr[1]:1, for i: 2 thru k do Arr[i]:0,
    for i: 2 thru n do (for j: k step -1 thru 1 do
      Arr[j]: j*Arr[j]+Arr[j-1]),tmp:Arr[k]+0);
(%o43) NS(k,n):=block ([Arr,tmp], array(Arr,if k>0 then k else 1), Arr_0:0,
  Arr_1:1, for i from 2 thru k do Arr_i:0, for i from 2 thru n do for j fr
    k step -1 thru 1 do Arr_j:j*Arr_j+Arr_{j-1}, tmp:Arr_k+0)
```

Откуда

```
(%i44) NS(4, 10); NS(7, 15);
(%o44) 34105
(%o45) 408741333
```

Ответ: а) $S(10, 4) = 34105$, б) $S(15, 7) = 408741333$.

Задача 2.7. Написать программу для вычисления чисел c_n Каталана. Найти: а) c_{10} , б) c_{15} .

Решение. Рекуррентное соотношение $c_n = \frac{4n-2}{n+1} c_{n-1}$, $c_0 = 1$ для чисел Каталана лишь коэффициентом отличается от соотношения для числа перестановок (задача 2.1). Поэтому функция для их вычисления может быть записана так:

```
(%i16) C(n):=block([c],c:1,for i:1 thru n do (c:((4*i-2)/(i+1))*c),c);
(%o16) C(n):=block([c],c:1,for i thru n do c:frac(4*i-2,i+1)*c,c)
```

Вычисления при $n = 10$, $n = 15$ дают:

```
(%i17) C(10);C(15);
(%o17) 16796
(%o18) 9694845
```

Ответ: а) $c_{10} = 16796$, б) $c_{15} = 9694845$.

Метод рекуррентных соотношений с успехом используется при решении задач на разбиение чисел на меньшие слагаемые. Типичными примерами таких задач являются задачи о размене и оплате. В пособии [2] (пункт 2.1.3) в качестве примера рассмотрена следующая задача.

Задача 2.8. Для оплаты некоторой коммунальной услуги надо перечислить через банкомат сумму в 2100 руб. Сколькими способами можно оплатить услугу банкнотами в 100, 500 и 1000 руб., если два способа, отличающиеся порядком, в котором банкноты отправляются в банкомат, считаются: а) различными, б) неразличимыми?

В пособии показано, что в общем случае число $f(n)$ способов разбиения целого неотрицательного числа n на натуральные слагаемые n_1, n_2, \dots, n_k , среди которых допускаются одинаковые слагаемые и порядок слагаемых является существенным, удовлетворяет рекуррентному соотношению

$$f(n) = f(n - n_1) + \dots + f(n - n_k),$$

где $f(n) = 1$, если $n = 1$, и $f(n) = 0$, если $n < 0$. В случае, когда порядок слагаемых является несущественным, число $F(n; n_1, n_2, \dots, n_k)$ способов разбиения числа n на слагаемые n_1, n_2, \dots, n_k удовлетворяет рекуррентному соотношению

$F(n; n_1, \dots, n_k) = F(n; n_1, \dots, n_{k-1}) + F(n - n_k; n_1, \dots, n_k)$,
при этом $F(0; n_1, \dots, n_k) = 1$.

Использование этих рекуррентных соотношений при решении задач с конкретными числовыми данными, как вы видели, сопряжено с громоздкими вычислениями. Поэтому дадим программные реализации соотношений, позволяющие избавиться от этих рутинных и скучных вычислений в подобных задачах.

Решение. Прежде всего заметим, что в первом случае рекуррентная последовательность задается с помощью k предыдущих членов, а во втором – лишь двух, но зависящих более чем от одной переменной.

а) Здесь необходимо использовать массив, так как в общем случае не известно, чему равно k . Кроме того, можно задавать массив коэффициентов при сложении предыдущих членов, формула вычисления текущего члена последовательности в котором будет выглядеть следующим образом:

$$f(n) = c_1 f(n - n_1) + \dots + c_k f(n - n_k).$$

Наконец, требуется задать начальные значения $f(1), f(2), \dots, f(\max(n_1, n_2, \dots, n_k))$. Ясно, что $f(j) = 0$ при $j < 0$ и $f(0) = 1$. Тогда начальные значения $f(1), f(2), \dots, f(\max(n_1, n_2, \dots, n_k))$ можно вычислить по той же формуле $f(n) = c_1 f(n - n_1) + \dots + c_k f(n - n_k)$.

С учетом сказанного имеем:

```
(%i1) RA1(CA, lenCA, n) :=
  block (
    [ft, k],
    array(FIarr, lenCA),
    FIarr[0]:1,
    k:0,
    for i: 1 thru lenCA do (
      if is(CA[i-1]#0) then k:i,
      if is(k=0) then FIarr[i]:0
      else (
        FIarr[i]:0,0
        for j:1 thru k do FIarr[i]:FIarr[i]+CA[j-1]*FIarr[i-j])),
    for i: lenCA+1 thru n do (
      ft: 0,
      for j: 1 thru lenCA-1 do (
        ft: ft+FIarr[j]*CA[lenCA-j], FIarr[j]: FIarr[j+1]),
      ft: ft+FIarr[lenCA]*CA[0], FIarr[lenCA]: ft), ft);
(%o1) RA1(CA, lenCA, n):=block ([ft, k], array(FIarr, lenCA), FIarr_0:1, k:0, for
i thru lenCA do (if is(CA_{i-1}#0) then k:i ,if is(k=0) then FIarr_i:0 else
(FIarr_i:0, for j thru k do FIarr_i:FIarr_i+CA_{j-1} FIarr_{i-j})), for i from lenCA
1 thru n do (ft:0, for j thru lenCA-1 do
(ft:ft+FIarr_j CA_{lenCA-j}, FIarr_j:FIarr_{j+1}), ft:ft+FIarr_{lenCA} CA_0, FIarr_{lenCA}:ft), i
)
```

Здесь блок

```
--> for i: 1 thru lenCA do (  
    if is(CA[i-1]#0) then k:i,  
    if is(k=0) then FIarr[i]:0  
    else (  
        FIarr[i]:0,  
        for j:1 thru k do FIarr[i]:FIarr[i]+CA[j-1]*FIarr[i-j])),
```

отвечает за нахождение начальных значений, а блок

```
--> for i: lenCA+1 thru n do (  
    ft: 0,  
    for j: 1 thru lenCA-1 do (  
        ft: ft+FIarr[j]*CA[lenCA-j],  
        FIarr[j]: FIarr[j+1]),  
    ft: ft+FIarr[lenCA]*CA[0],  
    FIarr[lenCA]: ft),
```

– за нахождение всех остальных членов последовательности. При этом входящие параметры функции – CA – массив коэффициентов при $f(n - n_j)$, заданный в порядке их следования.

Заметим, что задача о нахождении числа способов разложения числа $n = 2100$ на слагаемые $n_1 = 100$, $n_2 = 500$ и $n_3 = 1000$ равносильна задаче о нахождении числа способов разложения числа $n = 21$ на слагаемые $n_1 = 1$, $n_2 = 5$ и $n_3 = 10$. Поэтому, если используются 1-й, 5-й и 10-й элементы, то массив CA будет иметь вид $(1, 0, 0, 0, 1, 0, 0, 0, 0, 1)$, $lenCA$ – количество элементов массива, причем если в массиве CA содержится 10 элементов, пронумерованных от 0 до 9, то $lenCA=10$, а n – номер члена последовательности $f(j)$, который надо найти.

Запуская программу для численных данных, указанных в задаче, получим:

```
(%i28) array(CA, 9);  
CA[0]:1;CA[1]:0;CA[2]:0;CA[3]:0;CA[4]:1;CA[5]:0;  
CA[6]:0;CA[7]:0;CA[8]:0;CA[9]:1;listarray(CA);  
RA1(CA, 10, 21);  
  
(%o28) CA  
(%o29) 1  
(%o30) 0  
(%o31) 0  
(%o32) 0
```

```

(%o33) 1
(%o34) 0
(%o35) 0
(%o36) 0
(%o37) 0
(%o38) 1
(%o39) [1, 0, 0, 0, 1, 0, 0, 0, 0, 1]
(%o40) 268

```

б) В случае, когда порядок слагаемых не учитывается, функция для вычислений будет выглядеть так:

```

(%i2) RecF(n,IA,k):=
  block ([tmp,tmp1,res],if ( n<0 ) then (res: 0)
  else (if k=1 then (if is(mod(n,IA[0])=0) then res: 1 else res: 0)
  else (tmp: RecF (n,IA,k-1),tmp1: RecF (n-IA[k-1],IA,k),
  res: tmp+tmp1)),res);
(%o2) RecF(n,IA,k):=block ([tmp,tmp1,res],if n<0 then res:0 else if k=1
  then if is(mod(n,IA_0)=0) then res:1 else res:0 else
  (tmp:RecF(n,IA,k-1),tmp1:RecF(n-IA_{k-1},IA,k),res:tmp+tmp1),res)

```

Рассмотрим, как построена данная функция. Теперь порядок представления слагаемых разложения не важен. Первое граничное условие `if k=1 then (if is(mod(n,IA[0])=0) then res: 1 else res: 0)` отражает то, что количество способов представления слагаемого равно 0 или 1. Второе граничное условие показывает, что число меньшее нуля можно представить 0 способами (как следует из постановки задачи). Далее напрямую реализуется рекуррентное соотношение

$$F(n; n_1, \dots, n_k) = F(n; n_1, \dots, n_{k-1}) + F(n - n_k; n_1, \dots, n_k).$$

Запуская функцию для $n = 21$ и $n_1 = 1, n_2 = 5, n_3 = 10$, имеем:

```

(%i3) array(a,2); a[0]:1;a[1]:5;a[2]:10;
      RecF(21,a,3);
(%o3) a
(%o4) 1
(%o5) 5
(%o6) 10
(%o7) 9

```

Ответ: а) 268, б) 9.

Не менее эффективен метод рекуррентных соотношений в комбинаторных задачах на перечисление числа маршрутов заданной длины через вершины многоугольников, многогранников и графов.

Задача 2.9. Найти число замкнутых маршрутов длины n по сторонам равностороннего треугольника ABC со стороной 1. (Начальная и конечная точка маршрута – вершина A .) Провести вычисления для: а) $n = 15$, б) $n = 20$.

Решение. Пусть a_n – число маршрутов длины n из вершины A в вершину A , b_n – число маршрутов длины n из вершины A в вершину B и c_n – число маршрутов длины n из вершины A в вершину C .

Очевидно, что $a_0 = 1$, $a_1 = 0$ и числа a_n , b_n , c_n связаны системой рекуррентных соотношений

$$\begin{cases} c_n = b_n, \\ a_{n+1} = b_n + c_n, \\ b_{n+1} = a_n + c_n, \\ c_{n+1} = a_n + b_n. \end{cases}$$

Эта система равносильна системе

$$\begin{cases} a_{n+1} = 2b_n, \\ b_{n+1} = a_n + b_n. \end{cases}$$

Исключая из нее b_n , находим рекуррентное соотношение, связывающее числа a_n :

$$a_{n+2} = a_{n+1} + 2a_n, \quad a_0 = 1, \quad a_1 = 0.$$

Найденное рекуррентное соотношение отличается от соотношения Фибоначчи лишь постоянным коэффициентом 2 при a_n и начальными условиями $a_0 = 1$, $a_1 = 0$. Поэтому функцию для перечисления числа маршрутов длины n из вершины A в вершину A можно записать так:

```
(%i48) A(n):=block ( [at,a0,a1],a0:1,a1:0,
    for i: 2 thru n do (at: a1+2*a0,a0: a1,a1: at),at);
(%o48) A(n):=block
([at,a0,a1],a0:1,a1:0,for i from 2 thru n do (at:a1+2*a0,a0:a1,a1:at),a
```

Откуда при $n = 15$, $n = 12$ получим:

```
(%i49) A(15);A(20);
(%o49) 10922
(%o50) 349526
```

Ответ: а) $a_{15} = 10922$, б) $a_{20} = 349526$.

2. Рассмотрим примеры использования пакета **solve_rec** для решения рекуррентных соотношений. Этот пакет расширений загружается с помощью команды **load(solve_rec)\$**.

Задача 2.11. Найти общее решение линейного однородного рекуррентного соотношения:

а) $u_{n+5} = 3u_{n+4} + 5u_{n+3} - 27u_{n+2} + 32u_{n+1} - 12u_n$,

б) $u_{n+2} = -4u_n$,

в) $u_{n+4} = 4u_{n+3} - 8u_{n+2} + 8u_{n+1} - 4u_n$.

Решение.

а) Характеристическое уравнение

$$\lambda^5 - 3\lambda^4 - 5\lambda^3 + 27\lambda^2 + 32\lambda - 12 = 0$$

соотношения – уравнение 5-й степени. Хотя соотношение составлено так, что корни характеристического уравнения являются «хорошими» ($\lambda_{1,2} = 1, \lambda_{3,4} = 2, \lambda_5 = -3$), процесс их вычисления «вручную» достаточно трудоемок. Внося же соотношение в строку ввода, сразу находим:

```
(%i3) load(solve_rec)$
      solve_rec(u[n+5]=3*u[n+4]+5*u[n+3]-27*u[n+2]+32*u[n+1]-12*u[n],u[n]);
```

```
(%o4) u_n=(%k_5 n+%k_4)2^n+%k_1(-3)^n+%k_3 n+%k_2
```

Ответ: $u_n = k_1(-3)^n + k_2 + nk_3 + (k_4 + nk_5)2^n, n \in N_0$.

б) Характеристическое уравнение $\lambda^2 + 4 = 0$ соотношения с действительными коэффициентами имеет комплексно сопряженные корни $2i$ и $-2i$. Поскольку $2i = 2\left(\cos\frac{\pi}{2} + i\sin\frac{\pi}{2}\right)$, $-2i = 2\left(\cos\frac{\pi}{2} - i\sin\frac{\pi}{2}\right)$, то общее решение соотношения имеет вид: $u(n) = 2^n\left(C_1\cos\frac{n\pi}{2} + C_2\sin\frac{n\pi}{2}\right), n \in N_0$. Вычисления в Maxima выглядят следующим образом:

```
(%i6) load(solve_rec)$
      solve_rec(u[n+2]=-4*u[n],u[n]);
```

```
(%o7) u_n=%k_2(-1)^{n/2}2^n+(-%i)^n%k_12^n
```

Система «выдала» результат в комплексном виде и при этом произвела не слишком удачное «упрощение» степени ($i^n = (-1)^{n/2}$). Чтобы записать найденное решение без использования комплексного основания степени, пробуем представить его в тригонометрической форме с помощью функции **trigrat**:

```
(%i9) trigrat(%o7);
```

```
(%o9) u_n = %i (%k_2 - %k_1) 2^n sin(π n / 2) + (%k_2 + %k_1) 2^n cos(π n / 2)
```

Maxima, как видим, с задачей не справилась. Однако требуемый вид решения можно получить, если выполнить замену произвольных постоянных: $C_1 = (k_2 + k_1)$, $C_2 = i(k_2 - k_1)$.

Ответ: $u(n) = 2^n \left(C_1 \cos \frac{n\pi}{2} + C_2 \sin \frac{n\pi}{2} \right)$, $n \in N_0$.

в) Здесь характеристическое уравнение соотношения имеет два двукратных комплексно сопряженных корня

$$1 + i = 2 \left(\cos \frac{\pi}{4} + i \sin \frac{\pi}{4} \right), \quad 1 - i = 2 \left(\cos \frac{\pi}{4} - i \sin \frac{\pi}{4} \right).$$

Вычисления в Maxima выглядят следующим образом:

```
(%i13) load(solve_rec)$
solve_rec(u[n+4]=4*u[n+3]-8*u[n+2]+8*u[n+1]-4*u[n], u[n]);
```

```
(%o14) u_n = (%i + 1)^n (%k_4 n + %k_3) + (1 - %i)^n (%k_2 n + %k_1)
```

```
(%i15) trigrat(%o14);
```

```
(%o15) u_n = %i ((%k_4 - %k_2) n + %k_3 - %k_1) 2^(n/2) sin(π n / 4) + ((%k_4 + %k_2) n + %k_3 + %k_1) 2^(n/2) cos(π n / 4)
```

Вновь вводя новые обозначения

$$C_1 = \frac{k_3 + k_1}{2}, \quad C_2 = \frac{k_4 + k_2}{2}, \quad C_3 = \frac{i(k_3 - k_1)}{2}, \quad C_4 = \frac{i(k_4 - k_2)}{2},$$

ответ можно записать в привычном виде.

Ответ: $u(n) = 2^n (C_1 + nC_2) \cos \frac{n\pi}{4} + 2^n (C_3 + nC_4) \sin \frac{n\pi}{4}$, $n \in N_0$.

Задача 2.12. Найти общее решение соотношения Фибоначчи $F_{n+2} = F_{n+1} + F_n$ и частное решение, соответствующее начальным условиям $F(0) = 1$, $F(1) = 1$ (формулу Бине).

Решение. Данный простой пример выбран для того, чтобы показать, что система даже в хорошо известных случаях может выдавать результаты несколько в непривычном виде. Как известно, искомые решения имеют соответственно вид:

$$F(n) = k_1 \left(\frac{1+\sqrt{5}}{2} \right)^n + k_2 \left(\frac{1-\sqrt{5}}{2} \right)^n,$$

$$F_n = \frac{1}{\sqrt{5}} \left(\left(\frac{1+\sqrt{5}}{2} \right)^{n+1} - \left(\frac{1-\sqrt{5}}{2} \right)^{n+1} \right), \quad n \in N_0.$$

Посмотрим, как справиться с первой задачей система Maxima:

```
(%i9) load(solve_rec) $
      solve_rec(F[n+2]=F[n+1]+F[n], F[n]);
```

$$(\%o10) \quad F_n = \frac{(\sqrt{5}-1)^n k_1 (-1)^n + (\sqrt{5}+1)^n k_2}{2^n}$$

Здесь с точностью до обозначений результат тот же самый.

Перейдем ко второй задаче. Используя команду **Solve Linear System**, решаем систему, соответствующую начальным условиям $F(0) = 1, F(1) = 1$:

$$k_1 + k_2 = 1, \quad -\frac{\sqrt{5}-1}{2}k_1 + \frac{\sqrt{5}+1}{2}k_2 = 1:$$

```
(%i30) linsolve([k[1]+k[2]=1, -((sqrt(5)-1)/2)*k[1]+((sqrt(5)+1)/2)*k[2]=1],
               [k[1], k[2]]);
```

$$(\%o30) \quad [k_1 = -\frac{\sqrt{5}-5}{10}, k_2 = \frac{\sqrt{5}+5}{10}]$$

Подставляя найденные значения k_1, k_2 в общее решение, получим:

```
(%i32) ((sqrt(5)-1)^n*k1*(-1)^n)/2^n+((sqrt(5)+1)^n*k2)/2^n, k1=- (sqrt(5)-5)/10,
      k2=(sqrt(5)+5)/10;
```

$$(\%o32) \quad \frac{(5-\sqrt{5})(\sqrt{5}-1)^n 2^{-n-1} (-1)^n}{5} + \frac{(\sqrt{5}+1)^n (\sqrt{5}+5) 2^{-n-1}}{5}$$

Попытаемся выполнить преобразование полученного результата с помощью команд окна **Упростить**. Команда **Simplify radical**, позволяющая упрощать выражения с радикалами и степенными функциями с рациональными показателями, дает:

```
(%i33) radcan(%);
```

$$(\%o33) \quad \frac{2^{-n-1} \left(((\sqrt{5}-1)^n \sqrt{5}-5(\sqrt{5}-1)^n) (-1)^n + (-\sqrt{5}-5)(\sqrt{5}+1)^n \right)}{5}$$

Найденный ответ, однако, после упрощения «вручную» трудно привести к привычному виду:

$$\begin{aligned} &= \frac{2^{-n-1} \left(((\sqrt{5}-1)^n \sqrt{5}-5(\sqrt{5}-1)^n) (-1)^n + (-\sqrt{5}-5)(\sqrt{5}+1)^n \right)}{5} = \\ &= \frac{-\sqrt{5}(1-\sqrt{5})^n (1-\sqrt{5}) + \sqrt{5}(1+\sqrt{5})(1+\sqrt{5})^n}{5 \cdot 2^{n+1}} = \frac{1}{\sqrt{5}} \left(\left(\frac{1+\sqrt{5}}{2} \right)^{n+1} - \left(\frac{1-\sqrt{5}}{2} \right)^{n+1} \right). \end{aligned}$$

Задача 2.13. Найти частное решение линейного однородного рекуррентного соотношения $u_{n+3} = 3u_{n+2} - 2u_n$, если $u_0 = a$, $u_1 = b$, $u_2 = c$.

Решение. Находим общее решение соотношения:

```
(%i10) load(solve_rec)$
       solve_rec(u[n+3]=3*u[n+1]-2*u[n], u[n]);

(%o11)  $u_n = k_1 (-2)^n + k_2 n + k_3$ 
```

Из системы

$$\begin{cases} k_1 + k_2 = a, \\ -2k_1 + k_2 + k_3 = b, \\ 4k_1 + k_2 + k_3 = c \end{cases}$$

находим k_1, k_2, k_3 , соответствующие начальным условиям $u_0 = a, u_1 = b, u_2 = c$:

```
(%i12) linsolve([k1+k2=a, -2*k1+k2+k3=b,
                4*k1+k2+2*k3=c], [k1,k2,k3]);

(%o12)  $[k_1 = \frac{c-2b+a}{9}, k_2 = \frac{-c+2b+8a}{9}, k_3 = \frac{-c-b+2a}{3}]$ 
```

Подставляем найденные значения k_1, k_2, k_3 в общее решение:

```
(%i15) k1*(-2)^n+k3*n+k2, k1=(c-2*b+a)/9, k2=(-c+2*b+8*a)/9, k3=(-c-b+2*a)/3;

(%o15)  $\frac{(c-2b+a)(-2)^n}{9} + \frac{(c+b-2a)n}{3} + \frac{-c+2b+8a}{9}$ 
```

Ответ: $u_n = \frac{8a+2b-c}{9} + \frac{-2a+b+c}{3}n + \frac{a-2b+c}{9}(-2)^n, n \in N_0$.

Задача 2.14. Найти общее решение линейного неоднородного рекуррентного соотношения:

- а) $u_{n+2} = -2u_{n+1} + 8u_n + 3^n$;
- б) $u_{n+2} = -2u_{n+1} + 8u_n + 5n^2 - 3n + 1$;
- в) $u_{n+2} = -2u_{n+1} + 8u_n + \sin \frac{\pi n}{2}$.

Решение. Система Maxima прекрасно справляется с решением соотношений а) и б):

```
а)
(%i66) load(solve_rec)$
       solve_rec(u[n+2]=-2*u[n+1]+8*u[n]+3^n, u[n]);

(%o67)  $u_n = \frac{3^n}{7} + k_2 2^n + k_1 (-4)^n$ 
```

Ответ: $u_n = k_1(-4)^n + k_2 2^n + \frac{3^n}{7}$.

б)

```
(%i68) load(solve_rec)$
      solve_rec (u[n+2]=-2*u[n+1]+8*u[n]+5*n^2-3*n+1, u[n]);

(%o69) u_n = %k_2 2^n + %k_1 (-4)^n - n^2 - n - 11/5
```

Ответ: $u_n = k_1(-4)^n + k_2 2^n - n^2 - n - \frac{11}{5}$.

В случае в), однако, Maxima заявляет об ошибке:

```
(%i59) load (solve_rec)$
      solve_rec (u[n+2]=-2*u[n+1]+8*u[n]+sin((%pi*n)/2), u[n]);

(%o60) false
```

Поэтому поступим следующим образом. Согласно формуле Эйлера $e^{\frac{\pi ni}{2}} = \cos \frac{\pi n}{2} + i \sin \frac{\pi n}{2}$. Рассмотрим рекуррентное соотношение

$u_{n+2} = -2u_{n+1} + 8u_n + e^{\frac{\pi ni}{2}}$ или

$$u_{n+2} = -2u_{n+1} + 8u_n + \left(e^{\frac{\pi i}{2}}\right)^n.$$

Мнимая часть частного решения этого соотношения будет удовлетворять исходному рекуррентному соотношению. Находим:

```
(%i61) load (solve_rec)$
      solve_rec (u[n+2]=-2*u[n+1]+8*u[n]+(exp((%pi*i)/2))^n, u[n]);

(%o62) u_n = %k_2 2^n + %k_1 (-4)^n + \frac{(-1)^{n/2}}{2 %i - 9}
```

С помощью команды **Complex Simplification (Get Imaginary Part)** выделим мнимую часть дроби $\frac{(-1)^{n/2}}{-9+2i}$:

```
(%i68) imagpart ((-1)^(n/2)/(2*%i-9));

(%o68) -\frac{9 \sin\left(\frac{\pi n}{2}\right)}{85} - \frac{2 \cos\left(\frac{\pi n}{2}\right)}{85}
```

Следовательно, общее решение рекуррентного соотношения запишется так:

$$f(n) = C_1(-4)^n + C_2 2^n - \frac{1}{85} \left(2 \cos \frac{\pi n}{2} + 9 \sin \frac{\pi n}{2} \right), n \in N_0.$$

Уже на основании рассмотренных примеров можно заключить, что при решении линейных рекуррентных соотношений с постоянными коэффициентами система Maxima в одних случаях прекрасно справляется с поставленной задачей, в других – «берет на себя» рутинную часть вычислений, но полученный результат тре-

бует дальнейшей корректировки, в третьих – может быть хорошей помощницей, если пользователь располагает необходимой теоретической базой по решению данного вида соотношений.

Заметим также, что с помощью пакета **solve_rec** можно находить решения некоторых видов рекуррентных соотношений с данными функциональными коэффициентами, которые обычно тяжело решаются методом последовательных подстановок.

Задача 2.15. Найти решение рекуррентного соотношения

$$D_{n+1} = (n+1)D_n + (-1)^{n+1}, D_0 = 1.$$

Решение. Нетрудно видеть, что дано рекуррентное соотношение для числа D_n беспорядков, а требуется найти явную формулу для его вычисления. Это линейное соотношение имеет функциональные коэффициенты.

Применение команды **solve_rec** дает:

```
(%i21) load(solve_rec)$
      solve_rec(D[n+1]=(n+1)*D[n]+(-1)^(n+1), D[n], D[0]=1);

(%o22) 
$$D_n = n! - \sum_{j=0}^{n-1} \frac{(-1)^j}{(j+1)!} n!$$

```

Полученный ответ легко привести к привычному виду:

$$\begin{aligned} D_n &= n! - n! \sum_{j=0}^{n-1} \frac{(-1)^j}{(j+1)!} = n! \left(1 + \sum_{j=0}^{n-1} \frac{(-1)^{j+1}}{(j+1)!} \right) = \\ &= n! \left(1 + \sum_{j=1}^n \frac{(-1)^j}{j!} \right) = n! \left(\frac{1}{0!} + \sum_{j=1}^n \frac{(-1)^j}{j!} \right) = n! \sum_{j=0}^n \frac{(-1)^j}{j!} \end{aligned}$$

Ответ: $D_n = n! \sum_{j=0}^n \frac{(-1)^j}{j!}, n \in N_0.$

Задача 2.16. Найти общее решение рекуррентного соотношения $u_{n+1} = \frac{2n+1}{2n+3}u_n + \frac{2n-1}{2n+3}.$

Решение. Имеем:

```
(%i1) load("solve_rec")$

(%i2) solve_rec(u[n+1]=((2*n+1)/(2*n+3))*u[n]+((2*n-1)/(2*n+3)),u[n]);

(%o2) 
$$u_n = \frac{(n-2)n}{2n+1} + \frac{k_1}{2(n-1)+3}$$

```

Ответ: $u_n = \frac{k_1}{2(n-1)+3} + \frac{(n-2)n}{2n+1}, n \in N_0.$

2.3. Задачи для самостоятельного решения

Задачи 1–10 решить методом рекуррентных соотношений. При выполнении вычислений составить программу на встроенном макроязыке системы *Maxima*.

1. Имеется лента, разбитая на клетки и уходящая вправо в бесконечность. На первой клетке этой ленты сидит кузнечик. Из любой клетки кузнечик может перепрыгнуть либо на одну, либо на две клетки вправо. Сколькими способами он может добраться до n -й от начала ленты клетки? Провести вычисления для $n = 30$.

2. Лягушка прыгает по вершинам шестиугольника $ABCDEF$, перемещаясь каждый раз в одну из соседних вершин. Сколькими способами она может попасть из вершины A в вершину A за n прыжков? Провести вычисления для $n = 30$.

3. Лягушка прыгает по вершинам шестиугольника $ABCDEF$, перемещаясь каждый раз в одну из соседних вершин. Сколькими способами она может попасть из вершины A в вершину C за n прыжков? Провести вычисления для $n = 30$.

4. Лягушка прыгает по вершинам шестиугольника $ABCDEF$, перемещаясь каждый раз в одну из соседних вершин. Сколькими способами она может попасть из вершины A в вершину C за n прыжков при условии, что в вершину D ей прыгать нельзя? Провести вычисления для $n = 30$.

5. Сколькими способами купюру в 500 рублей можно разменять монетами в 1, 2, 5 и 10 рублей?

6. Сколькими способами можно разбить число N на слагаемые, каждое из которых равно одному из чисел $1, 2, \dots, n$? (Порядок слагаемых не учитывается.) Провести вычисления для $N = 30$, $n = 20$.

7. Сколькими способами можно разбить число N на различные слагаемые, каждое из которых равно одному из чисел $1, 2, \dots, n$? (Порядок слагаемых не учитывается.) Провести вычисления для $N = 30$, $n = 20$.

8. Имеются марки достоинством в n_1, n_2, \dots, n_k руб. (стоимости n_1, n_2, \dots, n_k марок различны и запас марок каждого вида неограничен). Сколькими способами из них можно выбрать марки на сумму в N руб. и затем наклеить на бандероль, если два способа наклейки, отличающиеся порядком наклейки марок, считаются

различными? Провести вычисления для $N = 30$, $n_1 = 1, n_2 = 2, n_3 = 3, n_4 = 5, n_5 = 10$.

9. Имеются марки достоинством в 1, 2, ..., n руб. Сколькими способами из них можно выбрать марки разного достоинства на сумму в N руб. и затем наклеить на бандероль, если два способа наклейки, отличающиеся порядком наклейки марок, считаются различными? Провести вычисления для $N = 30, n = 20$.

10. Найти число замкнутых маршрутов длины n по ребрам тетраэдра $ABCD$, если ребро тетраэдра равно 1. (Начальная и конечная точка маршрута – вершина A .) Провести вычисления для: $n = 12, n = 15$.

11. Найти общее решение линейного однородного рекуррентного соотношения:

а) $u_{n+4} = u_{n+3} + 7u_{n+2} - 13u_{n+1} + 6u_n$;

б) $u_{n+5} = u_{n+4} + 12u_{n+3} - 16u_{n+2} - 32u_{n+1} + 48u_n$;

в) $u_{n+5} = 15u_{n+3} - 10u_{n+2} - 60u_{n+1} + 72u_n$;

г) $u_{n+5} = -2u_{n+4} + 8u_{n+3} + 8u_{n+2} - 20u_{n+1} + 8u_n$;

д) $u_{n+4} = -2u_{n+3} + 4u_{n+2} + 2u_{n+1} - 3u_n$;

ж) $u_{n+5} = -2u_{n+4} - 4u_{n+3} + 8u_{n+2} + 16u_{n+1} + 32u_n$;

з) $u_{n+4} = -4u_{n+3} - 5u_{n+2} - 8u_{n+1} - 6u_n$;

и) $u_{n+4} = 3u_{n+3} - u_{n+2} - 3u_{n+1} + 2u_n$;

к) $u_{n+5} = u_{n+4} + 6u_{n+3} - 14u_{n+2} + 11u_{n+1} - 3u_n$.

12. Найти решение линейного однородного рекуррентного соотношения:

а) $u_{n+4} = -5u_{n+3} - 9u_{n+2} - 7u_{n+1} - 2u_n$,

$u_0 = 1, u_1 = -3, u_2 = 6, u_3 = -11$;

б) $u_{n+4} = 8u_{n+2} - 16u_n, u_0 = 2, u_1 = 0, u_2 = 24, u_3 = 0$;

в) $u_{n+3} = 3u_{n+2} - u_{n+1} + 3u_n, u_0 = 3, u_1 = 7, u_2 = 27$;

г) $u_{n+2} = 7u_{n+1} - 12u_n, u_0 = 2, u_1 = 4$;

д) $u_{n+2} = -u_{n+1} + 2u_n, u_0 = 4, u_1 = -2$;

е) $u_{n+3} = u_{n+2} + u_{n+1} - u_n, u_0 = 3, u_1 = 4, u_2 = 9$;

ж) $u_{n+2} = -3u_{n+1} - 2u_n, u_0 = -1, u_1 = 0$;

з) $u_{n+2} = 4u_{n+1} - 4u_n, u_0 = 2, u_1 = 6$;

и) $u_{n+3} = -4u_{n+2} + u_{n+1} + 4u_n, u_0 = 3, u_1 = -4, u_2 = 18$;

г) $u_{n+3} = 3u_{n+2} - 3u_{n+1} + u_n, u_0 = 0, u_1 = 0, u_2 = -2$;

к) $u_{n+2} = 2u_{n+1} - u_n, u_0 = -3, u_1 = -1$.

13. Найти общее решение линейного неоднородного рекуррентного соотношения:

а) $u_{n+2} = 5u_{n+1} - 6u_n + 2 \cdot 3^{n-1}$;

б) $u_{n+3} = -u_{n+1} + n$;

в) $u_{n+2} = 2u_{n+1} - 5u_n + 4n + 4$;

д) $u_{n+3} = 6u_{n+2} - 12u_{n+1} + 8u_n + 2^n$;

г) $u_{n+2} = 4u_n + 3n^2 - n$;

е) $u_{n+2} = -6u_n + \cos \frac{\pi n}{2}$;

ж) $u_{n+3} = u_{n+1} + n + 2^n$;

з) $u_{n+3} = 6u_{n+2} - 32u_n + \cos \frac{\pi n}{4}$;

и) $u_{n+2} = -6u_{n+1} - 9u_n + (-3)^{n-2}$;

к) $u_{n+2} = 4u_{n+1} - 3u_n + n^2 - 5n + 6$.

14. Найти решение линейного неоднородного рекуррентного соотношения:

а) $u_{n+2} = 3u_{n+1} + 2u_n + (-1)^n$, $u_0 = 1$, $u_1 = 2$;

б) $u_{n+3} = u_{n+2} + 4u_{n+1} - 4u_n + (n+1)2^n$,

$u_0 = 0$, $u_1 = 2$, $u_2 = 4$;

в) $u_{n+2} = 4u_{n+1} - 4u_n + 3^n$, $u_0 = 5$, $u_1 = 7$;

г) $u_{n+4} = 5u_{n+3} - 6u_{n+2} - 4u_{n+1} + 8u_n + (n+1)$,

$u_0 = 1$, $u_1 = 1$, $u_2 = 1$, $u_3 = 1$;

д) $u_{n+2} = -4u_{n+1} - 3u_n + 8k^2 + 4k + 10$, $u_0 = 1$, $u_1 = 1$;

е) $u_{n+2} = -u_{n+1} + \frac{1}{4}u_n + \left(\frac{1}{2}\right)^n$, $u_0 = 1$, $u_1 = \frac{3}{2}$;

ж) $u_{n+2} = -4u_{n+1} - 4u_n + 9k^2 + 3k + 11$, $u_0 = 0$, $u_1 = 1$;

з) $u_{n+3} = 3u_{n+1} - 2u_n + 3^n + 2^n$, $u_0 = 1$, $u_1 = 1$, $u_2 = 1$;

и) $u_{n+1} = u_n + n + 1$, $u_0 = 1$;

к) $u_{n+3} = 3u_{n+1} - 2u_n + n2^n$, $u_0 = -1$, $u_1 = 0$, $u_2 = 1$.

2.5. Варианты индивидуальных заданий

Номер варианта	Задачи
1.	1, 11 (а), 12 (а), 13 (а), 14 (а)
2.	2, 11 (б), 12 (б), 13 (б), 14 (б)
3.	3, 11 (в), 12 (в), 13 (в), 14 (в)
4.	4, 11 (г), 12 (г), 13 (г), 14 (г)
5.	5, 11 (д), 12 (д), 13 (д), 14 (д)

6.	6, 11 (е), 12 (е), 13 (е), 14 (е)
7.	7, 11 (а), 12 (а), 13 (а), 14 (а)
8.	8, 11 (ж), 12 (ж), 13 (ж), 14 (ж)
9.	9, 11 (и), 12 (и), 13 (и), 14 (и)
10.	10, 11 (к), 12 (к), 13 (к), 14 (к)

ЛАБОРАТОРНАЯ РАБОТА № 3

Производящие функции

В лабораторной работе № 3 рассматривается использование системы *Maxima* при решении перечислительных задач методом производящих функций. Даны примеры решения этим методом задач на выборки [2, § 3.4], разбиения конечных множеств [2, § 3.5] и разбиения натуральных чисел на слагаемые [2, § 3.6].

3.1. Теоретическая справка

Производящие функции. Один из наиболее универсальных методов решения перечислительных задач основан на использовании так называемых производящих функций. Основная идея этого метода состоит в сопоставлении последовательности комбинаторных чисел формального степенного ряда, коэффициенты которого являются членами этой последовательности. Такое соответствие позволяет свести процесс вычисления числа заданных комбинаторных конфигураций к операциям со степенными рядами, которые, как правило, оказываются логически более прозрачными, чем непосредственные комбинаторные рассуждения.

Формальным степенным рядом над полем \mathbf{R} действительных (над полем \mathbf{C} комплексных) чисел называется выражение

$$A(x) = a_0 + a_1x + \dots + a_nx^n + \dots \quad (3.1)$$

или, кратко,

$$A(x) = \sum_{n=0}^{\infty} a_n x^n,$$

где коэффициенты $a_n \in \mathbf{R}$, $n \in \mathbf{N}_0$ и x – некоторый произвольный символ, который принято называть переменной. Множество формальных степенных рядов над \mathbf{R} обозначают $\mathbf{R}[[x]]$.

Формальный степенной ряд

$$B(x) = b_0 + b_1x + \dots + b_nx^n + \dots \quad (3.2)$$

и ряд (3.1) считаются *равными*, если $a_n = b_n$ для любого $n \in \mathbf{N}_0$. Если все коэффициенты ряда (3.1), за исключением конечного числа, равны нулю, то $A(x)$ – многочлен над кольцом \mathbf{R} . Поэтому множество многочленов $\mathbf{R}[x]$ – подмножество множества $\mathbf{R}[[x]]$.

На множестве $\mathbf{R}[[x]]$ определены операции сложения и умножения формальных рядов, а также операцию умножения формального ряда на действительное число.

Суммой $A(x) + B(x)$ формальных рядов (3.1) и (3.2) называется ряд

$$A(x) + B(x) = \sum_{n=0}^{\infty} (a_n + b_n)x^n. \quad (3.3)$$

Нейтральным элементом относительно сложения рядов является нулевой ряд, т. е. ряд, все коэффициенты которого равны нулю: $A(x) + 0 = 0 + A(x) = A(x)$. Для любого ряда (3.1) существует противоположный ему ряд

$$-A(x) = -a_0 - a_1x - \dots - a_nx^n - \dots$$

такой, что $A(x) + (-A(x)) = (-A(x)) + A(x) = 0$.

Произведением формального ряда $A(x)$ на число $k \in \mathbf{R}$ называется ряд

$$kA(x) = k \sum_{n=0}^{\infty} a_nx^n = \sum_{n=0}^{\infty} (ka_n)x^n. \quad (3.4)$$

Произведением Коши $A(x)B(x)$ формальных рядов (3.1) и (3.2) называется ряд $C(x) = c_0 + c_1x + \dots + c_nx^n + \dots$ такой, что

$$c_n = \sum_{k=0}^n a_k b_{n-k}. \quad (3.5)$$

Нейтральным элементом относительно умножения является ряд $1 = 1 + 0x + 0x^2 + \dots$.

Ряд $A^{-1}(x)$ такой, что $A(x)A^{-1}(x) = 1$, называется *обратным* к ряду $A(x)$. Для того чтобы формальный степенной ряд (3.1) над

\mathbf{R} был обратимым, необходимо и достаточно, чтобы его коэффициент $a_0 \neq 0$.

Для ряда (3.1) и обратимого ряда (3.2) существует единственный степенной ряд $C(x)$ такой, что $B(x)C(x) = A(x)$. Степенной ряд $C(x)$ называется *частным* от деления ряда $A(x)$ на ряд $B(x)$ и обозначается $\frac{A(x)}{B(x)}$.

Пусть даны ряд $A(y) = a_0 + a_1y + \dots + a_ny^n + \dots$ и ряд $B(x) = b_1x + \dots + b_nx^n + \dots$, коэффициент b_0 которого равен нулю. Формальный степенной ряд

$$A(B(x)) = a_0 + a_1b_1x + (a_1b_2 + a_2b_1^2)x^2 + \\ + (a_1b_3 + 2a_3b_1b_2 + a_2b_1^2)x^3 + \dots$$

называется *подстановкой* формального степенного ряда $B(x)$ в формальный степенной ряд $A(y)$.

Производной формального степенного ряда (3.1) над полем \mathbf{R} действительных (\mathbf{C} комплексных) чисел называется формальный степенной ряд

$$D(A(x)) = \left(\sum_{n=0}^{\infty} a_n x^n \right)' = \sum_{n=0}^{\infty} (n+1)a_{n+1}x^n. \quad (3.6)$$

Производная n -го порядка степенного ряда $A(x)$ определяется по индукции:

$$D^n(A(x)) = D(D^{n-1}(A(x))) \quad (3.7)$$

и обозначается $A^{(n)}(x)$.

Для формального степенного ряда $A(x)$ справедлив аналог формулы Маклорена:

$$A(x) = \sum_{n=0}^{\infty} \frac{A^{(n)}(0)}{n!} x^n, \quad (3.8)$$

где $A(0) = a_0$, $A^{(k)}(0) = k! a_k$ ($k = 1, 2, \dots$).

Интегралом формального степенного ряда (3.1) называется формальный степенной ряд

$$\int A(x) = \sum_{n=1}^{\infty} \frac{a_{n-1}}{n} x^n. \quad (3.9)$$

Наряду с обычными формальными степенными рядами над \mathbf{R} (над \mathbf{C}) вводят *экспоненциальные* формальные степенные ряды. Так называются выражения вида

$$A(x) = a_0 + a_1 \frac{x}{1!} + a_2 \frac{x^2}{2!} + \dots + a_n \frac{x^n}{n!} + \dots, \quad (3.10)$$

где коэффициенты $a_i \in \mathbf{R}$, $n \in \mathbf{N}_0$ и x – некоторый произвольный символ. Множество экспоненциальных формальных степенных рядов над \mathbf{R} обозначают $\mathbf{R}^*[[x]]$.

На множестве $\mathbf{R}^*[[x]]$ обычным образом определяются операции сложения рядов и умножения ряда на действительное число:

$$\begin{aligned} \sum_{n=0}^{\infty} a_n \frac{x^n}{n!} + \sum_{n=0}^{\infty} b_n \frac{x^n}{n!} &= \sum_{n=0}^{\infty} (a_n + b_n) \frac{x^n}{n!}, \\ k \sum_{n=0}^{\infty} a_n \frac{x^n}{n!} &= \sum_{n=0}^{\infty} k a_n \frac{x^n}{n!}. \end{aligned}$$

Однако произведение Коши экспоненциальных рядов не является экспоненциальным рядом. Вместо него на $\mathbf{R}^*[[x]]$ вводится произведение Блиссара. *Произведением Блиссара* экспоненциальных рядов $A(x) = \sum_{n=0}^{\infty} a_n \frac{x^n}{n!}$ и $B(x) = \sum_{n=0}^{\infty} b_n \frac{x^n}{n!}$ называется экспоненциальный ряд $C(x) = \sum_{n=0}^{\infty} c_n \frac{x^n}{n!}$ такой, что

$$c_n = \sum_{k=0}^n C_n^k a_k b_{n-k}. \quad (3.11)$$

Формальное дифференцирование и интегрирование экспоненциального ряда (3.10) дает:

$$\begin{aligned} A'(x) &= a_1 + a_2 \frac{x}{1!} + a_3 \frac{x^2}{2!} + \dots + a_{n+1} \frac{x^n}{n!} + \dots, \\ \int A(x) &= a_0 \frac{x}{1!} + a_1 \frac{x^2}{2!} + \dots + a_{n-1} \frac{x^n}{n!} + \dots. \end{aligned}$$

Пусть имеется некоторая последовательность действительных или комплексных чисел

$$a_0, a_1, a_2, \dots, a_n, \dots \quad (3.12)$$

Производящей функцией этой последовательности называется формальный степенной ряд

$$A(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n + \dots = \sum_{i=0}^{\infty} a_i x^i. \quad (3.13)$$

Экспоненциальной производящей функцией последовательности (3.1) называется формальный экспоненциальный ряд

$$A(x) = a_0 + a_1 \frac{x}{1!} + a_2 \frac{x^2}{2!} + \dots + a_n \frac{x^n}{n!} + \dots = \sum_{i=0}^{\infty} a_i \frac{x^i}{i!}. \quad (3.14)$$

При этом говорят, что производящие функции (3.13), (3.14) *генерируют (порождают)* последовательность (3.12).

Операции, определенные на множестве формальных степенных рядов, позволяют строить из данных производящих функций новые производящие функции.

Пусть даны последовательности

$$\langle a_n \rangle = \langle a_0, a_1, \dots, a_n, \dots \rangle, \langle b_n \rangle = \langle b_0, b_1, \dots, b_n, \dots \rangle.$$

Операции сложения и умножения на скаляр, примененные к их производящим функциям $A(x) = \sum_{n=0}^{\infty} a_n x^n$, $B(x) = \sum_{n=0}^{\infty} b_n x^n$, позволяют построить производящую функцию

$$\alpha A(x) + \beta B(x) = \sum_{n=0}^{\infty} (\alpha a_n + \beta b_n) x^n$$

последовательности $\langle \alpha a_n + \beta b_n \rangle$.

Произведение Коши $C(x) = A(x) B(x)$ является производящей функцией последовательности $\langle c_n \rangle = \langle \sum_{k=0}^n a_k b_{n-k} \rangle$, которая называется *сверткой Коши* последовательностей $\langle a_n \rangle$ и $\langle b_n \rangle$. Важный частный случай свертки Коши имеем при $A(x) = x^m$, $m \in \mathbf{N}$; свертка $x^m B(x) = \sum_{n=m}^{\infty} b_{n-m} x^n$ является производящей функцией последовательности $\langle 0, \dots, 0, b_0, b_1, \dots \rangle$, которая начинается с m нулей. Поэтому умножение производящей функции на x^m часто называют *операцией сдвига* (вправо на m позиций).

Если из производящей функции $B(x)$ вычесть ее первые m членов и найденную разность разделить на x^m , то найдем производящую функцию $\sum_{n=m}^{\infty} b_n x^{n-m}$ последовательности $\langle b_m, b_{m+1}, \dots \rangle$, полученной из $\langle b_n \rangle$ в результате отбрасывания m первых членов и сдвига остальных членов на m позиций влево.

Чтобы получить производящую функцию $\sum_{n=0}^{\infty} n a_n x^n$ последовательности $\langle n a_n \rangle$, нужно продифференцировать производящую функцию $A(x)$, а затем полученный результат умножить на x . Если же проинтегрировать производящую функцию $\frac{A(x) - a_0}{x}$, то получим производящую функцию $\sum_{n=1}^{\infty} \frac{a_n}{n} x^n$ последовательности $\langle \frac{a_n}{n} \rangle$.

Произведение Блиссара экспоненциальных производящих функций $A(x) = \sum_{n=0}^{\infty} a_n \frac{x^n}{n!}$, $B(x) = \sum_{n=0}^{\infty} b_n \frac{x^n}{n!}$ последовательностей $\langle a_n \rangle$, $\langle b_n \rangle$ является экспоненциальной производящей функцией последовательности $\langle c_n \rangle = \langle \sum_{k=0}^n C_n^k a_k b_{n-k} \rangle$, которая называется *сверткой Блиссара* исходных последовательностей.

Построенные с помощью степенных рядов, в частности многочленов, и операций над ними, производящие функции являются одним из наиболее универсальных средств решения самых разнообразных перечислительных комбинаторных задач.

В таблице 2 приведены наиболее часто встречающиеся при решении комбинаторных задач последовательности и их производящие функции.

Таблица 2

<i>Последовательность</i>	<i>Производящая функция</i>
$\langle 1, 1, 1, 1, \dots \rangle$	$\frac{1}{1-x} = \sum_{n=0}^{\infty} x^n$
$\langle 1, -1, 1, -1, 1, -1, \dots \rangle$	$\frac{1}{1+x} = \sum_{n=0}^{\infty} (-1)^n x^n$
$\langle 1, 0, \dots, 0, 1, 0, \dots, 0, 1, 0, \dots \rangle$	$\frac{1}{1-x^k} = \sum_{n=0}^{\infty} x^{kn}$
$\langle 1, a, a^2, a^3, \dots \rangle$	$\frac{1}{1-ax} = \sum_{n=0}^{\infty} a^n x^n$
$\langle 1, C_n^1, C_{n+1}^2, C_{n+2}^3, \dots \rangle$	$\frac{1}{(1-x)^n} = \sum_{k=0}^{\infty} C_{n+k-1}^k x^k$
$\langle 1, \frac{1}{1!}, \frac{1}{2!}, \frac{1}{3!}, \dots \rangle$	$e^x = \sum_{n=0}^{\infty} \frac{1}{n!} x^n$
$\langle 1, \frac{m}{1!}, \frac{m^2}{2!}, \frac{m^3}{3!}, \dots \rangle$	$e^{mx} = \sum_{n=0}^{\infty} \frac{m^n}{n!} x^n$

$\langle 1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \dots \rangle$	$\ln \frac{1}{1-x} = \sum_{n=1}^{\infty} \frac{1}{n} x^n$
$\langle 1, -\frac{1}{2}, \frac{1}{3}, -\frac{1}{4}, \dots \rangle$	$\ln(1+x) = \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n} x^n$

Выборки. Производящей функцией для числа сочетаний C_n^k из n элементов по k является многочлен

$$F(x) = (1+x)^n = \sum_{k=0}^n C_n^k x^k. \quad (3.15)$$

Многочлен

$$F(x) = (1+x+\dots+x^k)^n. \quad (3.16)$$

служит производящей функцией для числа сочетаний из n элементов с повторениями, если число повторений элемента не превышает k . В том случае, когда на число повторений не накладывается никаких ограничений, производящая функция является степенным рядом

$$\frac{1}{(1-x)^n} = \sum_{k=0}^{\infty} C_{n+k-1}^k x^k = \sum_{n=0}^{\infty} \bar{C}_n^k x^k. \quad (3.17)$$

При рассмотрении неупорядоченных выборок из совокупности элементов m типов a_1, a_2, \dots, a_m можно выделить два основных случая:

- 1) число элементов каждого типа является неограниченным;
- 2) число элементов каждого типа ограничено, т. е. выбор идет из мультимножества $A = \{n_1 a_1, n_2 a_2, \dots, n_m a_m\}$.

На выборки из разных совокупностей, в свою очередь, могут накладываться некоторые дополнительные условия. Поэтому производящие функции $F(x)$, с помощью которых изучаются эти выборки, имеют специфические особенности. При составлении производящей функции $F(x)$ для вычисления числа n -выборок определенного вида в первом случае руководствуются следующими общими правилами:

1°. Число неупорядоченных n -выборок из совокупности элементов m типов, в которых представлено хотя бы по одному элементу каждого типа, равно коэффициенту при x^n в произведении

$$F(x) = \underbrace{(x + x^2 + \dots) \dots (x + x^2 + \dots)}_{m \text{ раз}}.$$

2°. В общем случае число неупорядоченных n -выборок из совокупности элементов a_1, a_2, \dots, a_m равно коэффициенту при x^n в произведении

$$F(x) = F_1(x)F_2(x) \dots F_m(x), \quad (3.18)$$

где $F_i(x)$ – многочлен или ряд, который строится с учетом ограничений, наложенных на выбор элемента a_i . Именно, если из совокупности элементов a_i , $1 \leq i \leq m$ необходимо выбрать:

а) ровно k элементов, то эту совокупность в произведении (3.18) следует представить множителем x^k ;

б) не менее k_1 , но не более k_2 элементов, то эту совокупность в произведении (3.18) следует представить многочленом $(x^{k_1} + \dots + x^{k_2})$;

в) нечетное количество элементов, то эту совокупность представляет множитель $(x + x^3 + \dots + x^{2k+1} + \dots)$;

г) четное количество элементов, то эту совокупность представляет множитель $(1 + x^2 + \dots + x^{2k} + \dots)$;

д) количество элементов, кратное r , то эту совокупность следует представить множителем $(1 + x^r + \dots + x^{rk} + \dots)$ (понятно, что правило является обобщением предыдущего).

Правила 1°–2° можно применять и для подсчета числа неупорядоченных выборок в том случае, когда задано количество элементов каждого типа. Если, например, из мультимножества $A = \{n_1 a_1, n_2 a_2, \dots, n_m a_m\}$ необходимо выбрать $n \leq n_1 + \dots + n_m$ элементов, то производящей функцией для числа таких выборок является произведение многочленов

$$F(x) = (1 + x + \dots + x^{n_1}) \dots (1 + x + \dots + x^{n_m}).$$

Использование экспоненциальных производящих функций позволяет перечислять упорядоченные выборы. Действительно,

$$F^*(x) = (1 + x)^n = \sum_{k=0}^n \frac{n!}{(n-k)! k!} x^k = \sum_{k=0}^n A_n^k \frac{x^k}{k!}$$

– экспоненциальная производящая функция для числа размещений без повторений.

Экспоненциальной производящей функцией для числа $\bar{A}_n^k = n^k$ размещений из n элементов по k с повторениями является функция $F(x) = e^{nx}$. В самом деле,

$$F^*(x) = e^{nx} = \sum_{k=0}^n \frac{(nx)^k}{k!} = \sum_{k=0}^n n^k \frac{x^k}{k!}.$$

Рассмотрим упорядоченные n -выборки из совокупности элементов m типов a_1, a_2, \dots, a_m . Пусть для каждого $1 \leq i \leq m$ выбирается n_i элементов a_i и $n_1 + \dots + n_m = n$. Если количество элементов любого типа является неограниченным, то выборка элементов a_i может быть представлена членом $\frac{x^{n_i}}{n_i!}$ ряда $1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots$, а вся n -выборка – произведением

$$\left(1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots\right) \dots \left(1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots\right), \quad (3.19)$$

содержащим m сомножителей. Если это произведение представить в виде экспоненциального ряда, то коэффициент $\frac{n!}{n_1!n_2!\dots n_m!}$ при $\frac{x^n}{n!}$, где $x^{n_1}x^{n_2}\dots x^{n_m} = x^{n_1+n_2+\dots+n_m}$, даст число способов, которыми можно выбрать и упорядочить n элементов (по n_i элементов типа a_i).

Понятно, что в конкретной перечислительной задаче экспоненциальные ряды, представляющие в произведении (3.19) элементы типа a_i , вновь будут иметь специфические особенности, обусловленные условиями задачи. Поэтому при составлении экспоненциальной производящей функции следует руководствоваться теми же правилами 1° и 2°, которые сформулированы для обычных производящих функций.

Разбиения множеств. Приведем производящие функции для основных видов разбиений множеств.

1°. Количество k^n способов, которыми n -множество можно разбить на k попарно непересекающихся упорядоченных блоков, среди которых допускаются пустые блоки, ищется с помощью производящей экспоненциальной функции

$$F^*(x) = e^{kx}, \quad (3.20)$$

коэффициент при $\frac{x^n}{n!}$ у которой равен k^n .

2°. Количество способов, которыми n -множество можно разбить на k попарно непересекающихся упорядоченных блоков, среди которых нет пустых, позволяет найти производящая экспоненциальная функция

$$F^*(x) = (e^x - 1)^k. \quad (3.21)$$

3°. Число всех способов неупорядоченного разбиения n -множества на k непустых подмножеств, т. е. число $S(n, k)$ Стирлинга 2-го рода, можно найти с помощью экспоненциальной производящей функции

$$F_k^*(x) = \frac{1}{k!} (e^x - 1)^k = \sum_{n=k}^{\infty} S(n, k) \frac{x^n}{n!}. \quad (3.22)$$

Это число можно найти и с помощью обычной производящей функции

$$F_k(x) = \frac{x^k}{(1-x)(1-2x) \dots (1-kx)} = \sum_{n=k}^{\infty} S(n, k) x^n. \quad (3.23)$$

4°. Число всех неупорядоченных разбиений n -множества на непустые подмножества, т. е. число Белла $B(n)$, ищется с помощью экспоненциальной производящей функции

$$e^{e^x - 1} = \sum_{n=0}^{\infty} B(n) \frac{x^n}{n!}. \quad (3.24)$$

Попутно отметим, что число неупорядоченных разбиений n -множества в k циклах или, что тоже самое, число подстановок степени n , в разложении которых ровно k циклов, есть число $C(n, k)$ Стирлинга первого рода без знака. Для чисел $s(n, k)$ Стирлинга первого рода

$$F_k(x) = (x)_n = x(x-1) \dots (x-n+1)$$

— обычная производящая функция, а

$$F_n^*(x) = \frac{1}{k!} \ln^k(1+x)$$

— экспоненциальная.

Разбиения чисел. Рассмотрим различные виды разложений натурального числа n на слагаемые, или, что то же самое, раскладок n неразличимых предметов в неразличимые и неразличимые ящики. Начнем с разложений, порядок слагаемых в которых является несущественным.

1°. Производящая функция для числа разложений натурального числа n на различные натуральные слагаемые, равные n_1, n_2, \dots, n_k , порядок которых не учитывается, равна

$$F(x) = (1+x^{n_1})(1+x^{n_2}) \dots (1+x^{n_k}). \quad (3.25)$$

2°. Производящая функция для числа разложений натурального числа n на натуральные слагаемые, равные n_1, n_2, \dots, n_k , порядок которых не учитывается, имеет вид:

$$F(x) = (1 + x^{n_1} + x^{2n_1} + \dots)(1 + x^{n_2} + x^{2n_2} + \dots) \times \dots \times (1 + x^{n_k} + x^{2n_k} + \dots) = \frac{1}{(1-x^{n_1})(1-x^{n_2})\dots(1-x^{n_k})}. \quad (3.26)$$

Обратите внимание, что теперь не требуется, чтобы все слагаемые в разложении были различными. Иными словами, находится число целочисленных неотрицательных решений системы $n_1x_1 + n_2x_2 + \dots + n_kx_k = n$.

3°. Производящая функция для числа разбиений числа n на не более чем k слагаемых имеет вид

$$F(x) = \frac{1}{(1-x)(1-x^2) \dots (1-x^k)}, \quad (3.27)$$

а ровно на k слагаемых – вид:

$$F(x) = \frac{x^k}{(1-x)(1-x^2) \dots (1-x^k)}. \quad (3.28)$$

Когда слагаемые разбиения или их число не указаны, в качестве производящих функций приходится рассматривать произведения, содержащие бесконечное число сомножителей.

4°. Количество разложений числа n на различные слагаемые, которые не указаны и порядок которых не учитывается, можно представить функцией

$$F(x) = (1+x)(1+x^2) \dots (1+x^k) \dots \quad (3.29)$$

Доказывается, что коэффициент при x^k в $F(x)$ равен коэффициенту при x^k в многочлене

$$(1+x)(1+x^2) \dots (1+x^k).$$

Производящие функции для числа разложений на различные четные и различные нечетные слагаемые имеют соответственно вид:

$$F(x) = (1+x^2)(1+x^4) \dots (1+x^{2k}) \dots;$$

$$F(x) = (1+x)(1+x^3) \dots (1+x^{2k+1}) \dots.$$

5°. Производящие функции для числа разложений числа n на произвольные слагаемые, на произвольные четные и произвольные нечетные слагаемые имеют соответственно вид (порядок слагаемых не учитывается):

$$F(x) = \frac{1}{(1-x)(1-x^2)\dots(1-x^k)\dots}. \quad (3.30)$$

$$F(x) = \frac{1}{(1-x^2)(1-x^4)\dots(1-x^{2k})\dots},$$

$$F(x) = \frac{1}{(1-x)(1-x^3)\dots(1-x^{2k+1})\dots}.$$

В заключение приведем производящие функции для разложений, в которых порядок слагаемых считается существенным.

1°. Производящая функция

– для числа способов представления натурального числа n в виде суммы k целых неотрицательных слагаемых, порядок которых учитывается:

$$F(x) = \frac{1}{(1-x)^k};$$

– для числа способов представления натурального числа n в виде суммы k натуральных слагаемых, каждое из которых не меньше l и порядок которых учитывается:

$$F(x) = (x^l + x^{l+1} + \dots)^k = \frac{x^{kl}}{(1-x)^k};$$

– для числа способов представления натурального числа n в виде суммы k натуральных слагаемых, порядок которых учитывается:

$$F(x) = \frac{x^k}{(1-x)^k}.$$

2°. Количество способов, которыми натуральное число n можно представить в виде суммы k слагаемых, принимающих значения n_1, n_2, \dots, n_m (порядок слагаемых учитывается), представляется производящей функцией

$$F(x) = (x^{n_1} + x^{n_2} + \dots + x^{n_m})^k.$$

3.2. Примеры решения задач с использованием Maxima

Нельзя не видеть, что при решении задач с конкретными числовыми данными методом производящих функций приходится сталкиваться с малоприятной и весьма громоздкой процедурой перемножения многочленов и степенных рядов. Для выполнения подобных вычислений система Maxima просто незаменима. При этом в подавляющем большинстве случаев придется пользоваться всего двумя командами:

– **expand**, которая раскрывает скобки и, следовательно, позволяет перемножать многочлены.

– **taylor**, которая выполняет разложение функции в ряд Тейлора.

1. В этом пункте, используя систему Maxima, проведем вычисления в задачах, рассмотренных в учебном пособии [2].

Задача 3.1. В продаже в неограниченном количестве имеются яблоки, груши и апельсины. Сколькими способами можно произвести покупку:

- а) 9 фруктов;
- б) 9 фруктов, среди которых хотя бы по одному фрукту каждого вида;
- в) 9 фруктов, среди которых ровно 3 апельсина;
- г) 9 фруктов, среди которых не менее 3 апельсинов;
- д) 9 фруктов, среди которых не более 3 апельсинов;
- е) 9 фруктов, среди которых не менее 4 яблок, но не более 2 апельсинов;
- ж) 9 фруктов, из которых можно составить три одинаковых набора фруктов?

Решение. Для каждого пункта на основе правил 1^о–2^о составим соответствующие производящие функции:

а) $(1 + x + x^2 + \dots)^3 = \frac{1}{(1-x)^3};$

б) $(x + x^2 + \dots)^3 = x^3(1 + x + x^2 + \dots)^3 = \frac{x^3}{(1-x)^3};$

в) $x^3(1 + x + x^2 + \dots)^2 = \frac{x^3}{(1-x)^2};$

г) $(x^3 + x^4 + \dots)(1 + x + x^2 + \dots)^2 = \frac{x^3}{(1-x)^3};$

д) $(1 + x + x^2 + x^3)(1 + x + x^2 + \dots)^2 = \frac{1+x+x^2+x^3}{(1-x)^2};$

е) $(x^4 + x^5 + \dots)(1 + x + x^2)(1 + x + x^2 + \dots) =$
 $= \frac{(1+x+x^2)x^4}{(1-x)^2} = \frac{(1-x^3)x^4}{(1-x)^3}.$

ж) $(1 + x^3 + x^6 + \dots)^3 = \frac{1}{(1-x^3)^3}.$

Для ответов на поставленные вопросы достаточно разложить найденные функции в степенные ряды и найти коэффициенты при x^9 . Используем для этого функцию **taylor**.

а)

```
(%i1) taylor(1/(1-x)^3,x,0,9);  
(%o1)/T/ 1+3 x+6 x^2+10 x^3+15 x^4+21 x^5+28 x^6+36 x^7+45 x^8+55 x^9+...
```

б), г)

```
(%i2) taylor(x^3/(1-x)^3,x,0,9);  
(%o2)/T/ x^3+3 x^4+6 x^5+10 x^6+15 x^7+21 x^8+28 x^9+...
```

в)

```
(%i3) taylor(x^3/(1-x)^2,x,0,9);  
(%o3)/T/ x^3+2 x^4+3 x^5+4 x^6+5 x^7+6 x^8+7 x^9+...
```

д)

```
(%i4) taylor((1+x+x^2+x^3)/(1-x)^2,x,0,9);  
(%o4)/T/ 1+3 x+6 x^2+10 x^3+14 x^4+18 x^5+22 x^6+26 x^7+30 x^8+34 x^9+...
```

е)

```
(%i5) taylor(((1-x^3)*x^4)/(1-x)^3,x,0,9);  
(%o5)/T/ x^4+3 x^5+6 x^6+9 x^7+12 x^8+15 x^9+...
```

ж)

```
(%i6) taylor(1/(1-x^3)^3,x,0,9);  
(%o6)/T/ 1+3 x^3+6 x^6+10 x^9+...
```

Ответ: а) 55, б) 28, в) 7, г) 28, д) 34, е) 15, ж) 10.

Задача 3.2. В вазе лежит 5 яблок, 6 груш и 4 апельсина. Сколькими способами можно выбрать из них:

а) 9 фруктов;

б) 9 фруктов, среди которых хотя бы по одному фрукту каждого вида;

в) 9 фруктов, среди которых ровно 3 апельсина;

г) 9 фруктов, среди которых не менее 3 апельсинов;

д) 9 фруктов, среди которых не более 3 апельсинов;

е) 9 фруктов, среди которых не менее 4 яблок, но не более 2 апельсинов;

ж) 9 фруктов, из которых можно составить три одинаковых набора фруктов?

Решение. Для каждого пункта составим соответствующие производящие функции. При составлении необходимо учитывать, что число фруктов каждого вида ограничено, поэтому:

а) $(1 + x + \dots + x^5)(1 + x + \dots + x^6)(1 + x + \dots + x^4);$

б) $(x + \dots + x^5)(x + \dots + x^6)(x + \dots + x^4);$

в) $(1 + x + \dots + x^5)(1 + x + \dots + x^6)x^3;$

г) $(1 + x + \dots + x^5)(1 + x + \dots + x^6)(x^3 + x^4);$

д) $(1 + x + \dots + x^5)(1 + x + \dots + x^6)(1 + x + x^2 + x^3);$

е) $(x^4 + x^5)(1 + x + \dots + x^6)(1 + x + x^2);$

ж) $(1 + x^3)^2(1 + x^3 + x^6).$

В этой задаче для вычислений используется функция **expand**:

а)

```
(%i10) expand((1+x+x^2+x^3+x^4+x^5)*
(1+x+x^2+x^3+x^4+x^5+x^6)*(1+x+x^2+x^3+x^4));
(%o10) x^15+3 x^14+6 x^13+10 x^12+15 x^11+20 x^10+24 x^9+26 x^8+26 x^7+24 x^6+20 x^5+15 x^4+
10 x^3+6 x^2+3 x+1
```

б)

```
(%i11) expand((x+x^2+x^3+x^4+x^5)*
(x+x^2+x^3+x^4+x^5+x^6)*(x+x^2+x^3+x^4));
(%o11) x^15+3 x^14+6 x^13+10 x^12+14 x^11+17 x^10+18 x^9+17 x^8+14 x^7+10 x^6+6 x^5+3 x^4+x^3
```

в)

```
(%i12) expand((1+x+x^2+x^3+x^4+x^5)*
(1+x+x^2+x^3+x^4+x^5+x^6)*x^3);
(%o12) x^14+2 x^13+3 x^12+4 x^11+5 x^10+6 x^9+6 x^8+5 x^7+4 x^6+3 x^5+2 x^4+x^3
```

г)

```
(%i13) expand((1+x+x^2+x^3+x^4+x^5)*
(1+x+x^2+x^3+x^4+x^5+x^6)*(x^3+x^4));
(%o13) x^15+3 x^14+5 x^13+7 x^12+9 x^11+11 x^10+12 x^9+11 x^8+9 x^7+7 x^6+5 x^5+3 x^4+x^3
```

д)

```
(%i14) expand((1+x+x^2+x^3+x^4+x^5)*
(1+x+x^2+x^3+x^4+x^5+x^6)*(1+x+x^2+x^3));
(%o14) x^14+3 x^13+6 x^12+10 x^11+14 x^10+18 x^9+21 x^8+22 x^7+21 x^6+18 x^5+14 x^4+10 x^3+6
x^2+3 x+1
```

е)

```
(%i15) expand((x^4+x^5)*(1+x+x^2+x^3+x^4+x^5+x^6)*(1+x+x^2));
(%o15) x^13+3 x^12+5 x^11+6 x^10+6 x^9+6 x^8+6 x^7+5 x^6+3 x^5+x^4
```

ж)

```
(%i16) expand((1+x^3)^2*(1+x^3+x^6));
```

```
(%o16) x^12+3 x^9+4 x^6+3 x^3+1
```

Ответ: а) 24, б) 18, в) 6, г) 12, д) 18, е) 6, ж) 3.

Задача 3.3. Найти количество шестизначных чисел, все цифры в которых нечетные, а цифры 1 и 3 встречаются не менее одного раза.

Решение. Имеется 5 нечетных цифр. Поэтому экспоненциальная производящая функция, позволяющая найти количество указанных шестизначных чисел, запишется следующим образом:

$e^{3x} \left(\frac{x}{1!} + \frac{x^2}{2!} + \dots \right)^2 = e^{3x} (e^x - 1)^2$. Разложим ее в ряд Тейлора:

```
(%i18) taylor((%e^(3*x))*(%e^x-1)^2,x,0,6);
```

```
(%o18)/T/ x^2+4 x^3+97 x^4/12+11 x^5+4081 x^6/360+...
```

Поскольку для перечисления упорядоченных выборок используется экспоненциальная функция, найденный коэффициент при x^6 нужно умножить на $6!$:

```
(%i19) 6!*4081/360;
```

```
(%o19) 8162
```

Ответ: 8162.

Отсюда следует, что искомое количество шестизначных чисел равно $5^6 - 2 \cdot 4^6 + 3^6 = 8225$.

Задача 3.4. Имеется 5 карточек, на которых написана буква А, 4 карточки с буквой Б и 3 карточки с буквой В. Сколько различных «слов» можно составить с помощью этих карточек, в которых:

а) 6 букв;

б) 6 букв, среди которых хотя бы по одной букве каждого вида;

в) 6 букв, среди которых ровно 3 буквы А;

г) 6 букв, среди которых не менее 4 букв А;

д) 6 букв, среди которых не более 2 букв А;

е) 6 букв, среди которых четное число букв А, не менее 2 букв Б и не более 2 букв В?

Решение. Теперь производящие функции составляются с учетом того, что выбор каждой из букв ограничен:

$$\begin{aligned}
 \text{a)} & \left(1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!}\right) \cdot \left(1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!}\right) \cdot \\
 & \cdot \left(1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!}\right); \\
 \text{б)} & \left(\frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!}\right) \left(\frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!}\right) \left(\frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!}\right); \\
 \text{в)} & \frac{x^3}{3!} \left(\frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!}\right) \left(\frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!}\right); \\
 \text{г)} & \left(\frac{x^4}{4!} + \frac{x^5}{5!}\right) \left(1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!}\right) \left(1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!}\right); \\
 \text{д)} & \left(1 + \frac{x}{1!} + \frac{x^2}{2!}\right) \left(1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!}\right) \left(1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!}\right); \\
 \text{е)} & \left(1 + \frac{x^2}{2!} + \frac{x^4}{4!}\right) \left(\frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!}\right) \left(1 + \frac{x}{1!} + \frac{x^2}{2!}\right).
 \end{aligned}$$

Число искомых выборок будет равно произведению найденного коэффициента при x^6 на $6!$. Вычисления дают:

а)

```
(%i20) expand((1+x/1!+x^2/2!+x^3/3!+x^4/4!+x^5/5!)*(1+x/1!+x^2/2!+x^3/3!+x^4/4!)*
(1+x/1!+x^2/2!+x^3/3!));
```

```
(%o20)  $\frac{x^{12}}{17280} + \frac{x^{11}}{1440} + \frac{17 x^{10}}{3456} + \frac{11 x^9}{432} + \frac{33 x^8}{320} + \frac{121 x^7}{360} + \frac{107 x^6}{120} + \frac{77 x^5}{40} + \frac{10 x^4}{3} + \frac{9 x^3}{2} + \frac{9 x^2}{2} + 3 x + 1$ 
```

```
(%i21) 6!*107/120;
```

```
(%o21) 642
```

б)

```
(%i22) expand((x/1!+x^2/2!+x^3/3!+x^4/4!+x^5/5!)*(x/1!+x^2/2!+x^3/3!+x^4/4!)*
(x/1!+x^2/2!+x^3/3!));
```

```
(%o22)  $\frac{x^{12}}{17280} + \frac{x^{11}}{1440} + \frac{17 x^{10}}{3456} + \frac{217 x^9}{8640} + \frac{71 x^8}{720} + \frac{3 x^7}{10} + \frac{17 x^6}{24} + \frac{5 x^5}{4} + \frac{3 x^4}{2} + x^3$ 
```

```
(%i23) 6!*17/24;
```

```
(%o23) 510
```

в)

```
(%i24) expand(x^3/3!*(x/1!+x^2/2!+x^3/3!+x^4/4!)*
(x/1!+x^2/2!+x^3/3!));
```

```
(%o24)  $\frac{x^{10}}{864} + \frac{7 x^9}{864} + \frac{5 x^8}{144} + \frac{7 x^7}{72} + \frac{x^6}{6} + \frac{x^5}{6}$ 
```

```
(%i25) 6!/6;
```

```
(%o25) 120
```

г)

```
(%i26) expand((x^4/4!+x^5/5!)*(1+x/1!+x^2/2!+x^3/3!+x^4/4!)*
(1+x/1!+x^2/2!+x^3/3!));
```

```
(%o26) 
$$\frac{x^{12}}{17280} + \frac{x^{11}}{1440} + \frac{13 x^{10}}{3456} + \frac{x^9}{72} + \frac{107 x^8}{2880} + \frac{13 x^7}{180} + \frac{x^6}{10} + \frac{11 x^5}{120} + \frac{x^4}{24}$$

```

```
(%i27) 6!/10;
```

```
(%o27) 72
```

д)

```
(%i28) expand((1+x/1!+x^2/2!)*(1+x/1!+x^2/2!+x^3/3!+x^4/4!)*
(1+x/1!+x^2/2!+x^3/3!));
```

```
(%o28) 
$$\frac{x^9}{288} + \frac{x^8}{32} + \frac{23 x^7}{144} + \frac{41 x^6}{72} + \frac{3 x^5}{2} + \frac{71 x^4}{24} + \frac{13 x^3}{3} + \frac{9 x^2}{2} + 3 x + 1$$

```

```
(%i29) 6!*41/72;
```

```
(%o29) 410
```

е)

```
(%i30) expand((1+x^2/2!+x^4/4!)*(x^2/2!+x^3/3!+x^4/4!)*(1+x/1!+x^2/2!));
```

```
(%o30) 
$$\frac{x^{10}}{1152} + \frac{x^9}{192} + \frac{17 x^8}{576} + \frac{13 x^7}{144} + \frac{13 x^6}{48} + \frac{11 x^5}{24} + \frac{17 x^4}{24} + \frac{2 x^3}{3} + \frac{x^2}{2}$$

```

```
(%i31) 6!*13/48;
```

```
(%o31) 195
```

Ответ: а) 642, б) 510, в) 120, г) 72, д) 410, е) 195.

2. Перейдем к задачам на разбиение множеств.

Задача 3.5. Найти число способов, которыми 7 различных открыток можно разложить в 5 конвертов так, чтобы не было пустых конвертов. Рассмотреть случаи: а) все конверты одинаковые, б) все конверты различные.

в) Сколькими способами 7 различных открыток можно разложить по конвертам так, чтобы не было пустых конвертов?

Решение. а) Очевидно, что нужно найти $S(7,5)$. Разложение функции $F_n^*(x) = \frac{1}{5!}(e^x - 1)^5$ в ряд Тейлора дает:

```
(%i2) taylor(1/5!*(%e^x-1)^5, x, 0, 7);
```

```
(%o2) /T/ 
$$\frac{x^5}{120} + \frac{x^6}{48} + \frac{x^7}{36} + \dots$$

```

Следовательно, $S(7,5) = \frac{7!}{36}$:

```
(%i3) 7!/36;  
(%o3) 140
```

Этот же результат получим, если используем обычную производящую функцию (3.23):

```
(%i8) taylor(x^5/((1-x)*(1-2*x)*(1-3*x)*(1-4*x)*(1-5*x)), x, 0, 7);  
(%o8)/T/ x^5+15 x^6+140 x^7+...
```

б) Нужно найти коэффициент при $\frac{x}{5!}$ в разложении функции $(e^x - 1)^5$. Имеем:

```
(%i9) taylor((%e^x-1)^5, x, 0, 7);  
(%o9)/T/ x^5+5 x^6/2+10 x^7/3+...  
(%i10) 7!*10/3;  
(%o10) 16800
```

в) Требуется найти число Белла $B(7)$. Вычисления дают:

```
(%i13) taylor(%e^(%e^x-1), x, 0, 7);  
(%o13)/T/ 1+x+x^2+5 x^3/6+5 x^4/8+13 x^5/30+203 x^6/720+877 x^7/5040+...  
(%i14) 7!*877/5040;  
(%o14) 877
```

Ответ: а) 140, б) 16800, 877.

Рассмотрим пример решения задачи на упорядоченные разбиения с ограничениями.

Задача 3.6. Сколькими способами 9 пассажиров могут сесть в 3 легковых автомобиля так, чтобы:

- а) в каждый автомобиль сел хотя бы один пассажир;
- б) в один автомобиль село нечетное число пассажиров, а в два других – четное;
- в) в каждый автомобиль село по три человека?

(Порядок рассадки пассажиров в автомобиле считать несущественным.)

Решение. Учитывая, что в автомобиль может сесть не более четырех пассажиров, составим соответствующие производящие функции:

$$a) \left(\frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} \right)^3;$$

$$б) \left(\frac{x}{1!} + \frac{x^3}{3!} \right) \left(\frac{x^2}{2!} + \frac{x^4}{4!} \right)^2;$$

$$в) \left(\frac{x^3}{3!} \right)^3 = \frac{x^9}{3!^3}.$$

Выполним вычисления:

а)

```
(%i3) expand((x/1!+x^2/2!+x^3/3!+x^4/4!)^3);
```

```
(%o3)  $\frac{x^{12}}{13824} + \frac{x^{11}}{1152} + \frac{7x^{10}}{1152} + \frac{53x^9}{1728} + \frac{11x^8}{96} + \frac{x^7}{3} + \frac{3x^6}{4} + \frac{5x^5}{4} + \frac{3x^4}{2} + x^3$ 
```

```
(%i4) 9!*53/1728;
```

```
(%o4) 11130
```

б)

```
(%i5) expand((x/1!+x^3/3!)*(x^2/2!+x^4/4!)^2);
```

```
(%o5)  $\frac{x^{11}}{3456} + \frac{5x^9}{576} + \frac{x^7}{12} + \frac{x^5}{4}$ 
```

```
(%i6) 9!*5/576;
```

```
(%o6) 3150
```

в)

```
(%i7) 9!/3!^3;
```

```
(%o7) 1680
```

Ответ: а) 11130, б) 3150, в) 1680.

Следующая задача демонстрирует случай, когда экспоненциальные производящие функции используются для перечисления разбиений, в которых упорядочены элементы в блоках разбиения.

Задача 3.7. Сколькими способами 8 пассажиров могут сесть в 3 легковых автомобиля, если:

а) допускается, что в один из автомобилей пассажиры не сели;

б) в каждый автомобиль должен сесть хотя бы один пассажир?

Решение. а) Число всех упорядоченных разбиений n -множества на k упорядоченных блоков равно

$$n! C_{n+k-1}^{k-1} = A_{n+k-1}^{k-1}.$$

Это число разбиений можно найти как коэффициент при $\frac{x^n}{n!}$ в экспоненциальной производящей функции

$$F_k^*(x) = \frac{1}{(1-x)^k} = \sum_{n=0}^{\infty} C_{n+k-1}^n x^n = \sum_{n=0}^{\infty} n! C_{n+k-1}^n \frac{x^n}{n!}.$$

В рассматриваемом случае:

```
(%i17) taylor(1/(1-x)^3, x, 0, 8);
(%o17) /T/ 1+3 x+6 x^2+10 x^3+15 x^4+21 x^5+28 x^6+36 x^7+45 x^8+...

(%i18) 8!*45;
(%o18) 1814400
```

б) Число всех упорядоченных разбиений n -множества на k упорядоченных блоков, каждый из которых не пуст, равно $n! C_{n-1}^{k-1}$. Здесь экспоненциальная производящая функция –

$$F_k^*(x) = \frac{x^k}{(1-x)^k} = \sum_{n=0}^{\infty} C_{n+k-1}^n x^{n+k} = \sum_{n=k}^{\infty} C_{n-1}^{n-k} x^n = \sum_{n=k}^{\infty} n! C_{n-1}^{k-1} \frac{x^n}{n!}.$$

В нашем случае:

```
(%i19) taylor(x^3/(1-x)^3, x, 0, 8);
(%o19) /T/ x^3+3 x^4+6 x^5+10 x^6+15 x^7+21 x^8+...

(%i20) 8!*21;
(%o20) 846720
```

Ответ: а) 1814400, б) 846720.

Решим с помощью производящих функций для чисел $s(n, k)$ Стирлинга первого рода задачу 1.9 (б) из лабораторной работы № 1.

Задача 3.8. Найти число всех подстановок 12-й степени, в разложении которых ровно 5 циклов.

С помощью первой функции получаем:

```
(%i39) taylor(x*(x-1)*(x-2)*(x-3)*(x-4)*(x-5)*(x-6)*
(x-7)*(x-8)*(x-9)*(x-10)*(x-11), x, 0, 5);
(%o39) /T/ -39916800 x+120543840 x^2-150917976 x^3+105258076 x^4-45995730 x^5+...
```

Коэффициент при x^5 равен 45995730.

К этому же результату приходим, используя экспоненциальную производящую функцию:

```
(%i31) taylor(1/5!*(log(1+x))^5, x, 0, 12);
(%o31) /T/ x^5/120 - x^6/48 + 5 x^7/144 - 7 x^8/144 + 1069 x^9/17280 - 19 x^10/256 + 31063 x^11/362880 - 139381 x^12/1451520 + ...
```

В этом случае искомое число – коэффициент при $\frac{x^{12}}{12!}$.

```
(%i32) 12!*139381/1451520;
```

```
(%o32) 45995730
```

Ответ: 45995730.

3. В заключение рассмотрим задачи на разбиение чисел.

Задача 3.9. Сколько комплектов букетов можно составить из 15 одинаковых роз, если букеты могут состоять из 1, 3, 5, 7, 9 роз и число цветов во всех букетах комплекта должно быть различным?

Решение. Согласно (3.25) составляем функцию

$$F(x) = (1+x)(1+x^3)(1+x^5)(1+x^7)(1+x^9),$$

и перемножаем входящие в нее многочлены:

```
(%i4) expand((1+x)*(1+x^3)*(1+x^5)*(1+x^7)*(1+x^9));
```

```
(%o4) x^25+x^24+x^22+x^21+x^20+x^19+x^18+2 x^17+2 x^16+2 x^15+x^14+2 x^13+2 x^12+x^11+2 x^10+
2 x^9+2 x^8+x^7+x^6+x^5+x^4+x^3+x+1
```

Ответ: 2.

Задача 3.10. Сколькими способами купюру в 1000 рублей можно разменять на купюры в 50, 100 и 500 рублей?

Решение. Теперь не требуется, чтобы все слагаемые в разложении были различными, поэтому для решения применяется производящая функция (3.26). Очевидно, что данная задача эквивалентна задаче о разложении числа 20 на слагаемые, равные 1, 2 и 10. Составим производящую функцию

$$F(x) = \frac{1}{(1-x)(1-x^2)(1-x^{10})}.$$

Находим ее разложение в ряд Тейлора:

```
(%i5) taylor(1/((1-x)*(1-x^2)*(1-x^10)),x,0,20);
```

```
(%o5) 1/T/ 1+x+2 x^2+2 x^3+3 x^4+3 x^5+4 x^6+4 x^7+5 x^8+5 x^9+7 x^10+7 x^11+9 x^12+9 x^13+11
x^14+11 x^15+13 x^16+13 x^17+15 x^18+15 x^19+18 x^20+...
```

Ответ: 18.

Задача 3.10. Сколькими способами число 12 можно представить в виде суммы:

- различных слагаемых;
- различных четных слагаемых;
- различных нечетных слагаемых.

Порядок слагаемых не учитывается.

Решение. Производящие функции, соответствующие этим суммам, имеют вид:

$$\begin{aligned} \text{а)} & (1+x)(1+x^2)(1+x^3)(1+x^4)(1+x^5)(1+x^6) \cdot \\ & \cdot (1+x^7)(1+x^8)(1+x^9)(1+x^{10})(1+x^{11})(1+x^{12}); \\ \text{б)} & (1+x^2)(1+x^4)(1+x^6)(1+x^8)(1+x^{10})(1+x^{12}); \\ \text{в)} & (1+x)(1+x^3)(1+x^5)(1+x^7)(1+x^9)(1+x^{11}). \end{aligned}$$

Раскрывая скобки, получим:

а)

```
(%i6) expand((1+x)*(1+x^2)*(1+x^3)*(1+x^4)*(1+x^5)*(1+x^6)*
(1+x^7)*(1+x^8)*(1+x^9)*(1+x^10)*(1+x^11)*(1+x^12));
(%o6) x^78+x^77+x^76+2 x^75+2 x^74+3 x^73+4 x^72+5 x^71+6 x^70+8 x^69+10 x^68+12 x^67+15 x^66
+17 x^65+20 x^64+24 x^63+27 x^62+31 x^61+36 x^60+40 x^59+45 x^58+51 x^57+56 x^56+61 x^55+67
x^54+72 x^53+78 x^52+84 x^51+89 x^50+94 x^49+100 x^48+104 x^47+108 x^46+113 x^45+115 x^44+11
x^43+121 x^42+122 x^41+123 x^40+124 x^39+123 x^38+122 x^37+121 x^36+118 x^35+115 x^34+113
x^33+108 x^32+104 x^31+100 x^30+94 x^29+89 x^28+84 x^27+78 x^26+72 x^25+67 x^24+61 x^23+56
x^22+51 x^21+45 x^20+40 x^19+36 x^18+31 x^17+27 x^16+24 x^15+20 x^14+17 x^13+15 x^12+12 x^11+
10 x^10+8 x^9+6 x^8+5 x^7+4 x^6+3 x^5+2 x^4+2 x^3+x^2+x+1
```

б)

```
(%i8) expand((1+x^2)*(1+x^4)*(1+x^6)*(1+x^8)*(1+x^10)*(1+x^12));
(%o8) x^42+x^40+x^38+2 x^36+2 x^34+3 x^32+4 x^30+4 x^28+4 x^26+5 x^24+5 x^22+5 x^20+5 x^18+4
x^16+4 x^14+4 x^12+3 x^10+2 x^8+2 x^6+x^4+x^2+1
```

в)

```
(%i7) expand((1+x)*(1+x^3)*(1+x^5)*(1+x^7)*(1+x^9)*(1+x^11));
(%o7) x^36+x^35+x^33+x^32+x^31+x^30+x^29+2 x^28+2 x^27+2 x^26+2 x^25+3 x^24+2 x^23+2 x^22+3
x^21+3 x^20+3 x^19+2 x^18+3 x^17+3 x^16+3 x^15+2 x^14+2 x^13+3 x^12+2 x^11+2 x^10+2 x^9+2 x^8+;
+x^6+x^5+x^4+x^3+x+1
```

Ответ: а) 15, б) 4, в) 3.

Задача 3.11 (о счастливых билетах). Трамвайные билеты имеют шестизначные номера. Билет считается «счастливым», если сумма его первых трех цифр равна сумме трех последних. Найти число всех счастливых билетов.

Решение. Существует взаимно однозначное соответствие между множеством счастливых шестизначных билетов и множеством номеров с суммой цифр 27. Для того чтобы установить такое соответствие, заменим в произвольном счастливом билете три последние цифры его шестизначного номера на цифры, дополняющие их до 9 (например, номер 713425 при такой замене перейдет в номер 713574, сумма цифр которого равна 27). Если сумма трех

первых цифр (и трех последних) прежнего номера была равна k , то после указанной замены сумма трех последних цифр нового номера станет равной $27 - k$. Следовательно, общая сумма цифр нового номера равна 27.

Обратно, пусть в номере, сумма цифр которого равна 27, сумма трех первых цифр равна k , тогда сумма трех последних – $27 - k$. Заменяя в нем последние три цифры на цифры, дополняющие их до 9, получим «счастливый» номер.

Число способов, которыми число 27 можно представить в виде суммы 6 слагаемых, каждое из которых равно одному из чисел 0, 1, 2, ..., 9 (порядок слагаемых учитывается), можно найти с помощью производящей функции

$$F(x) = (1 + x + x^2 + \dots + x^9)^6.$$

Выполним возведение многочлена в степень:

```
(%i12) expand((1+x+x^2+x^3+x^4+x^5+x^6+x^7+x^8+x^9)^6);
(%o12) x^54+6 x^53+21 x^52+56 x^51+126 x^50+252 x^49+462 x^48+792 x^47+1287 x^46+2002 x^45+
2997 x^44+4332 x^43+6062 x^42+8232 x^41+10872 x^40+13992 x^39+17577 x^38+21582 x^37+2592
x^36+30492 x^35+35127 x^34+39662 x^33+43917 x^32+47712 x^31+50877 x^30+53262 x^29+54747
x^28+55252 x^27+54747 x^26+53262 x^25+50877 x^24+47712 x^23+43917 x^22+39662 x^21+35127
x^20+30492 x^19+25927 x^18+21582 x^17+17577 x^16+13992 x^15+10872 x^14+8232 x^13+6062 x^1
+4332 x^11+2997 x^10+2002 x^9+1287 x^8+792 x^7+462 x^6+252 x^5+126 x^4+56 x^3+21 x^2+6 x+
```

Коэффициент при x^{27} равен 55252.

Ответ: 55252.

Задача 3.12 (задача абитуриента). Поступающий в высшее учебное заведение должен сдать 3 экзамена. Он полагает, что проходной балл составит 13 баллов. Сколькими способами можно сдать экзамены, чтобы набрать проходной балл?

Решение. Проходной балл – 13 – разбивается на три слагаемых (число экзаменов), которые могут принимать значения 3, 4 и 5 (положительные отметки), причем порядок слагаемых имеет значение (экзамены разные). Поэтому

$$F(x) = (x^3 + x^4 + x^5)^3 = x^9(1 + x + x^2)^3 = x^9 \left(\frac{1-x^3}{1-x} \right)^3.$$

Разложим функцию в ряд Тейлора:

```
(%i13) taylor(x^9*(1-x^3)/(1-x)^3,x,0,13);
(%o13) /T/ x^9+3 x^10+6 x^11+7 x^12+6 x^13+...
```

Ответ: 6.

3.4. Задачи для самостоятельного решения

Решить задачи методом производящих функций.

1. В урне имеются 5 красных, 4 синих и 3 белых шара. Сколькими способами можно выбрать из урны 8 шаров? Во скольких случаях среди выбранных шаров будут 3 красных и 2 синих шара?

2. В урне имеются 5 красных, 4 синих и 3 белых шара. Из урны выбирается 8 шаров. Во скольких случаях среди выбранных шаров:

- а) хотя бы один шар будет белым;
- б) ровно один шар будет белым;
- в) будет четное число красных шаров и нечетное – синих?

3. В вазе лежат конфеты четырех сортов, по 4 конфеты каждого сорта. Сколькими способами сладкоежка можно съесть:

- а) 8 конфет;
- б) 8 конфет, среди которых хотя бы по одной конфете каждого сорта;
- в) 8 конфет, среди которых ровно 4 конфеты одного, особо понравившегося сорта?

4. В продаже имеются карандаши красного, синего, желтого, зеленого и черного цветов. Сколькими способами можно купить 20 карандашей так, чтобы

- а) количество карандашей красного цвета было не менее 5 и не более 10;
- б) количество карандашей красного цвета было кратно 5, а синего цвета – кратно 3, количество желтых карандашей было не более 3, зеленых – не менее 3 и черных – не более 2?

5. Сколько десятизначных чисел можно составить из цифр 1, 2, 3, 4, если в числе должно быть:

- а) 5 цифр 1;
- б) не менее одной цифры 1, двух цифр 2, нечетное число цифр 3 и четное число цифр 4;
- в) поровну четных и нечетных цифр?

6. Сколько десятизначных чисел можно составить из цифр 1, 2, 3, 4, 5, если в числе:

- а) было 2 цифры 1 и 3 цифры 2;

б) цифра 1 встречалась не менее 2, но не более 5 раз, цифра 2 встречалась 2 раза, а цифра 3 – 1 раз;

в) было нечетное число нечетных цифр?

7. Сколько десятизначных чисел можно составить из цифр 1, 2, 3? Сколько среди них делится на 9?

8. Переплетчик должен переплести 12 различных книг в красный, синий и коричневый переплеты. Сколькими способами он может это сделать, если в каждый цвет должна быть переплетена хотя бы одна книга?

9. Сколько слов можно составить из букв слова «логарифм», в которых две гласные буквы и не менее двух согласных?

10. Сколько различных наборов из 6 шаров можно составить, имея в наличии 2 одинаковых красных шара, 3 одинаковых синих шара и 4 одинаковых белых шара?

11. В лифт вошли 8 человек. Сколькими способами они могут выйти на четырех этажах так, чтобы на каждом этаже вышел по крайней мере один человек?

12. Сколькими способами можно разделить 15 различных предметов между тремя людьми так, чтобы каждый получил: а) не менее 2-х и не более 10 предметов, б) по 5 предметов.

13. Сколькими способами четверо детей могут разделить между собой 10 белых грибов, 12 подберезовиков и 9 подосиновиков? Сколькими способами можно разделить эти грибы так, чтобы каждый получил хотя бы 7 грибов?

14. Девушка собирается отправить своему приятелю по почте 8 своих различных фотографий, используя для этого 5 конвертов. Сколькими способами она это может сделать, если: а) конверты различные, б) конверты одинаковые?

15. В хоккейной команде 24 игрока. Сколькими способами их можно разместить в гостинице в шести четырехместных номерах?

16. Сколькими способами 3 юноши и 5 девушек могут разбиться на две команды по 4 человека так, чтобы в каждой команде был хотя бы один юноша?

17. Сколькими способами можно расставить 10 различных книг в книжном шкафу с пятью полками, если каждая полка может вместить все 10 книг?

18. Сколькими способами можно разделить на команды по 6 человек для игры в волейбол группу из: а) 12, б) 24 спортсменов?

19. Сколькими способами 9 постояльцев можно заселить в четыре номера гостиницы, если в номер нужно разместить не менее двух, но не более четырех человек?

20. Пять пиратов делят между собой добычу, состоящую из 18 различных драгоценных украшений. Сколькими способами может произойти раздел, если:

- а) он идет без каких-либо ограничений;
- б) каждый пират должен получить хотя бы одно украшение;
- в) точно два пирата останутся без добычи;
- г) четыре пирата получают по 4 украшения, а пятый – 2;
- д) каждый пират должен получить не менее трех украшений?

21. Найти число целочисленных неотрицательных решений уравнения $x_1 + 2x_2 + \dots + 5x_k = 10$.

22. Найти число способов, которыми число 30 можно представить в виде суммы 3 слагаемых, каждое из которых равно одному из чисел 0, 1, 2, ..., 15 (порядок слагаемых учитывается).

23. Сколькими способами можно оплатить покупку в 220 рублей купюрами в 100 и 50 рублей и монетами в 5 и 10 рублей?

24. Десять ящиков пронумерованы числами от 1 до 10. Сколькими способами можно разложить по этим ящикам 20 одинаковых шаров так, чтобы ни один ящик не оказался пустым?

25. Сколькими способами можно составить сумму в 25 рублей, если в наличии имеется 5 монет достоинством в 1 рубль, 4 монеты достоинством в 2 рубля, 3 монеты в 5 рублей и 2 монеты в 10 рублей?

26. Сколькими способами можно составить сумму в 25 рублей, если в наличии имеется по три монеты достоинством в 1, 2, 5 и 10 рублей?

27. Сколькими способами можно составить сумму в 25 рублей из монет достоинством в 1, 2, 5 и 10 рублей?

28. Сколькими способами число 20 можно разложить: а) на сумму слагаемых, б) на сумму четных слагаемых, в) на сумму нечетных слагаемых? (Порядок слагаемых не учитывается.)

29. Имеются рублевые, двухрублевые и десятирублевые марки. На конверт надо наклеить 6 марок на сумму 22 рубля. Сколькими

способами это можно сделать, если: а) учитывается порядок наклейки марок, б) порядок наклейки марок является несущественным?

30. Одновременно бросается 5 игральных костей. Во скольких случаях сумма выпавших очков окажется равной 21?

31. Найти число подстановок n -й степени, в разложении которых ровно k циклов.

Вариант	а	б	в	г	д	е	ж	з	и	к
n	8	10	6	7	5	9	8	10	6	7
k	3	7	1	4	4	5	5	3	5	3

3.5. Варианты индивидуальных заданий

Номер варианта	Задачи
1.	1, 11, 21, 31 (а)
2.	2, 12, 22, 31 (б)
3.	3, 13, 23, 31 (в)
4.	4, 14, 24, 31 (г)
5.	5, 15, 25, 31 (д)
6.	6, 16, 26, 31 (е)
7.	7, 17, 27, 31 (ж)
8.	8, 18, 28, 31 (з)
9.	9, 19, 29, 31 (и)
10.	10, 20, 30, 31 (к)

ВМЕСТО ЗАКЛЮЧЕНИЯ

Авторы пособия надеются, что выполнение лабораторных работ продемонстрировало читателю, что система компьютерной математики *Math* обладает хорошим набором функций для решения комбинаторных задач и хорошей скоростью выполнения вычислений. Надеемся также, что он убедился в том, что в качественных текстовых задачах *Math* может быть лишь помощником в выполнении необходимых численных вычислений, решение задач ему придется искать самому, опираясь на хорошее знание теоретического материала.

ЛИТЕРАТУРА

Основная

1. Виленкин Н. Я. Комбинаторика. – М.: Наука, 1969. – 328 с.
2. Клековкин Г. А. Введение в перечислительную комбинаторику. – М.: Спутник+, 2014. – 219 с.
3. Стахин Н. А. Основы работы с системой аналитических (символьных) вычислений Махіта (ПО для решения задач аналитических (символьных) вычислений): учебное пособие. – М.: 2008. – 86 с.
4. Чичкарёв Е. А. Компьютерная математика с Махіта: руководство для школьников и студентов. – М.: ALT Linux, 2009. – 233 с.

Дополнительная

5. Алфутова Н. Б. Алгебра и теория чисел: сборник задач для математических школ / Н. Б. Алфутова, А. В. Устинов. – 2-е изд., испр. и доп. – М.: МЦНМО, 2004. – 320 с.
6. Гаврилов Г. П. Задачи и упражнения по дискретной математике: учебное пособие / Г. П. Гаврилов, А. А. Сапоженко. – М.: ФИЗМАТЛИТ, 2004. – 416 с.
7. Кузьмин О. В. Перечислительная комбинаторика: учебное пособие. – М.: Дрофа, 2005. – 110 с.
8. Программирование в Махіта: учебное пособие / А. Р. Есаян, В. Н. Чубариков, Н. М. Добровольский, А. В. Якушин. – Тула: Изд-во Тульского гос. пед. ун-та им. Л. Н. Толстого, 2012. – 351 с.
9. Эвнин А. Ю. Сборник задач по дискретной математике. – 4-е изд. перераб. и доп. – М.: ЛИБРОКОМ, 2011. – 264 с.

Учебное издание

Клековкин Геннадий Анатольевич
Богданов Павел Сергеевич

РЕШЕНИЕ КОМБИНАТОРНЫХ ЗАДАЧ
С ИСПОЛЬЗОВАНИЕМ СРЕДЫ
MAXIMA

Учебное пособие

Компьютерная верстка и корректура
Н. В. Мыльникова

Обложка – ООО «Абрис»

Самарский филиал ГБОУ ВПО города Москвы
«Московский городской педагогический университет»,
443041, г. Самара, ул. Братьев Коростелевых, 76.

Подписано в печать 13.03.15. Формат 60х90 ¹/₁₆. Бумага офисная.
Гарнитура Times New Roman. Печать оперативная. Усл. печ. л. 5,5.

Тираж 200 экз. Заказ 1303.

Отпечатано в типографии ООО «Абрис»,
443041, г. Самара, ул. Агибалова, 78.