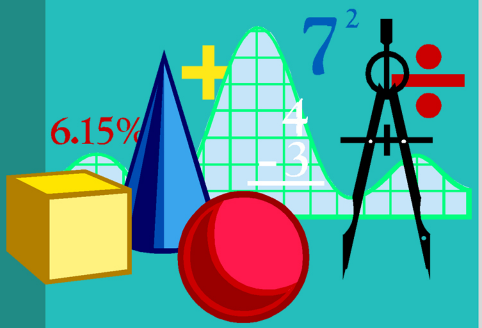
Министерство образования и науки Хабаровского края

Краевое государственное автономное образовательное учреждение дополнительного образования
«Центр развития творчества детей
(Региональный модельный центр дополнительного образования детей Хабаровского края)»

Центр технического творчества

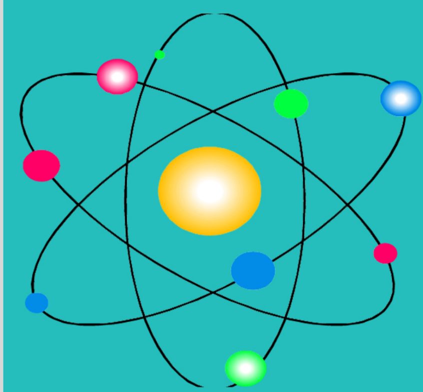
МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

Математика



Информатика

Физика



в образовательных организациях
Хабаровского края

МИФ

№ 4 (12) 2019 г.

г. Хабаровск
2019 г.

Печатается по решению
научно-методического совета
КГАОУ ДО РМЦ,
протокол № 1 от 26.03.2019 г.

МИФ: математика, информатика, физика в образовательных организациях Хабаровского края . Методические рекомендации / Сост. И.В. Воеводина. – Хабаровск: КГАОУ ДО РМЦ, 2019. – 44 с.

В методических рекомендациях представлены статьи по математике, информатике и физике.

Данный материал будет полезен студентам, педагогам физики общеобразовательных учреждений, педагогам, работающим по дополнительным общеобразовательным программам физико-математического направления.

Ответственный редактор: В.В. Шевченко
Ответственный за выпуск: М.Н. Никитенко
Компьютерная верстка: В.Д. Шабалдина

Научные консультанты:
по математике: Жулидова Ю.В., старший преподаватель кафедры математики и информационных технологий ФГБОУ ВО ТОГУ;
по физике: Горбанева Л.В., преподаватель кафедры физики ФГБОУ ВО ТОГУ;
по информатике: Редько Е.А., старший преподаватель кафедры математики и информационных технологий ФГБОУ ВО ТОГУ.

Содержание

Введение

Современное общество ждёт мыслящих, инициативных, творческих выпускников с широким кругозором и прочными знаниями. Образовательные учреждения в условиях модернизации системы образования ищут пути, которые позволили бы им выполнить этот заказ общества. Познавательная самостоятельность — социально значимое свойство личности. Её развитие относится к числу тех проблем, от успешного решения которых зависит научно-технический и социальный прогресс.

На разных этапах обучения нестандартные занятия обладают значительными формирующими возможностями и функциями. Организовать интересные, познавательные, нестандартные занятия по математике и физике педагогам помогут статьи Н.Е. Пишковой «Методы решения занимательных и логических задач» и Л.В. Горбаневой, А.М. Голобокова «Применение фокусов на занятиях по физике». Темы, раскрытые в данных статьях, не только расширяют границы знаний учащихся, но и формируют стойкий интерес к предметам физико-математического цикла.

Одним из важнейших показателей развития отечественного образования и работы с одарёнными детьми являются результаты международных предметных олимпиад. В настоящее время таких олимпиад семь. Среди них — Международная олимпиада по информатике (МОИ), которая занимает важное место в силу востребованности современных информационных технологий при подготовке высококвалифицированных кадров. Решение олимпиадных задач является одним из эффективных и проверенных на практике педагогических механизмов выявления и развития творческих способностей детей, важной составляющей профильного обучения, обеспечивающей высокую мотивацию к образовательной и научной деятельности. В статье Е.А. Редько «Решение олимпиадных задач по программированию» доступным языком рассмотрены решения задач и приведены задания для самостоятельной работы.

Представленные в методических рекомендациях статьи будут интересны как педагогам, так и учащимся.

МАТЕМАТИКА

Методы решения занимательных и логических задач

Н.Е. Пишкова,
преподаватель кафедры математики и информационных технологий
ФГБОУ ВО ПИ ТОГУ

Дискретная (конечная) математика — представляет собой важное направление в математике, изучающее дискретные математические структуры. К числу таких конечных структур могут быть отнесены, например, графы, утверждения в логике, а также некоторые математические модели преобразователей информации, конечные автоматы, машина Тьюринга и т. п.

Основные разделы дискретной математики: «Комбинаторный анализ», «Теория графов», «Теория кодирования» и некоторые другие.

В дискретной математике можно выделить характерные методы и задачи. Элементы дискретной математики возникли в глубокой древности и, развиваясь параллельно с другими разделами математики, в значительной мере являлись их составной частью. Типичными для того периода были задачи, связанные со свойствами целых чисел, которые затем привели к созданию теории чисел. К их числу могут быть отнесены найденные алгоритмы сложения и умножения натуральных чисел у древних египтян (II тыс. до н. э.), задачи о суммировании и вопросы делимости натуральных чисел в пифагорийской школе (VI в. до н. э.) и т. д. Позже (XVII–XVIII вв.), в основном в связи с игровыми задачами появились элементы комбинаторного анализа и дискретной теории вероятностей, а в связи с общими проблемами теории чисел, алгебры и геометрии (XVIII–XIX вв.) возникли важнейшие понятия алгебры, определившие развитие и содержание алгебры на много лет вперёд и имевшие, по существу, дискретную природу.

Стремление к строгости математических рассуждений и анализ рабочего инструмента математики — логики — привели к выделению ещё одного важного раздела математики — математической логики (XIX–XX вв.). Однако наибольшего развития дискретная математика достигла в связи с запросами практики, приведшими к появлению новой науки — кибернетики и её теоретической части — математической кибернетики (XX в.).

Дискретная математика стала активно развиваться с начала XX века, когда стали изучаться возможности формализации математики, и были получены фундаментальные результаты в области математической логики. Информатизация и компьютеризация общества во второй половине XX века в значительной степени стимулировала развитие дискретной математики.

Методы дискретной математики используются при решении как серьёзных задач (таких, как проектирование микросхем или программирование баз данных), так и при решении различных занимательных и логических задач. Рассмотрим несколько сюжетов, связанных с такими, весьма интересными задачами.

1. Логические задачи

Под логическими задачами обычно понимают такие задачи, которые решают с помощью одних лишь логических операций. Логические задачи могут решаться и фактически решаются обычными рассуждениями. Иногда их решение требует длинных рассуждений, необходимое направление которых заранее нельзя предугадать.

Такими задачами увлекались ещё в древности. Вот одна из них.

Задача 1. Утомившись от споров и летнего зноя, три древнегреческих философа прилегли под деревом сада Академии и уснули. Пока они спали, шутники испачкали углём их лбы. Проснувшись, и взглянув друг на друга, все пришли в весёлое настроение, начали смеяться, но это никого не тревожило, так как каждому казалось естественным, что двое других смеются друг над другом. Внезапно один из мудрецов (**А**) перестал смеяться, так как сообразил, что его собственный лоб тоже испачкан. Как он рассуждал?

Решение.

А рассуждал так: «Каждый из нас может думать, что его собственный лоб чистый. **Б** уверен, что его лицо чистое и смеётся над измазанным лбом **В**. Но если бы **Б** видел, что моё лицо чистое, то он удивился бы смеху **В**, так как в этом случае **В** смеялся бы без причины. Однако **Б** не удивлён, значит он может думать, что **В** смеётся надо мной. Следовательно, мой лоб чёрный!»

Задача 2. Один путешественник отправился на остров, населённый двумя племенами, о которых ему было известно, что в одном племени все высокие, а в другом — низкие, что члены одного из племён всегда говорят правду, а члены другого всегда врут, и что те и другие знают английский язык.

Высадившись на остров, путешественник встретил двух туземцев: высокого и низкого. Высокий, на заданный ему по-английски вопрос путешественника: «Всегда ли вы говорите правду?», — ответил: «Карра бум», а низенький сказал, что это значит «да».

Какому племени принадлежал каждый туземец?

Решение.

1) Пусть низкий — лжец, тогда «кара бум» означает «нет»; в таком случае высокий не принадлежит ни к племени лжецов (так как там все низкие, а он высокий), ни к племени правдивцев (так как он не всегда говорит правду). Но это невозможно, так как он должен принадлежать к одному из племён.

2) Пусть низкий — правдивец, тогда «карра бум» означает «да»; и тогда высокий — лжец.

Есть особый вид логических задач, решение которых удобно оформлять в виде таблицы.

Задача 3. Три брата (Иван, Дмитрий и Сергей) преподают различные дисциплины (химию, историю, биологию) в университетах Москвы, Санкт-Петербурга и Киева.

1) Иван работает не в Москве, а Дмитрий — не в Санкт-Петербурге.
2) Москвич преподаёт не историю.
3) Тот, кто работает в Санкт-Петербурге, преподаёт химию.
4) Дмитрий преподаёт не биологию.

Что и в каком университете преподаёт Сергей?

Решение.

При решении этой задачи целесообразно по ходу рассуждений заполнять нижеприведённую таблицу знаками «л», «и», в зависимости от того ложно или истинно высказывание, соответствующее данной клетке таблицы.

Заполним таблицу в соответствии с условиями.

Исходя из условий 1) и 4), можем заполнить три клетки. Дальше рассуждаем так: «Ввиду того, что Дмитрий работает не в Санкт-Петербурге (1), а согласно (3) тот, кто работает в Санкт-Петербурге, преподаёт химию, то Дмитрий преподаёт не химию. В клетку, соответствующую строке «Дмитрий» и столбцу «Химия» ставим «л».

Москва	Петербург	Киев		Химия	Биология	История
л	**и**	**л**	Иван	**и**	**л**	**л**
и	**л**	**л**	Сергей	**л**	**и**	**л**
л	**л**	**и**	Дмитрий	**л**	**л**	**и**

Из таблицы видно, что Дмитрий преподаёт историю. В соответствующую клетку ставим «и». А для всех остальных (Ивана и Сергея) — «л».

Согласно (2) москвич преподаёт не историю. Следовательно, Дмитрий работает не в Москве. Но так как Иван тоже не работает в Москве, то там работает Сергей. Занесём эти данные в таблицу. Иван работает в Санкт-Петербурге, так как Дмитрий — в Киеве. Следовательно, согласно (3) Иван преподаёт химию. А так как Дмитрий преподаёт историю, то Сергей преподаёт биологию.

В результате постепенного заполнения получится таблица (не поленитесь проделать заполнение самостоятельно по ходу рассуждений!).

Ответ: Сергей работает в Московском университете и преподаёт биологию.

Для сложной и запутанной задачи, мы получили красивое и простое решение.

Задача 4. Встретились три друга: скульптор Белов, скрипач Чернов и художник Рыжов. «Замечательно, что один из нас — блондин, другой — брюнет, а третий — рыжеволосый, но ни у одного нет волос того цвета, на который указывает фамилия», — заметил брюнет. «Ты прав», — сказал Белов.

Какой цвет волос у художника?

Решение.

Заполним таблицу, учитывая условия задачи. Получим следующее:

Фамилия	Цвет волос		
	блондин	брюнет	рыжий
Белов	-	-	+
Чернов	+	-	-
Рыжов	-	+	-

Ответ: из таблицы видно, что художник — брюнет.

Существуют задачи, в которых дано несколько высказываний о предметах и сказано, сколько из них верно и сколько верно лишь наполовину.

Задача 5. В салоне небольшого самолёта было 42 пассажира. Некоторые из них были москвичами, остальные — иногородними. Среди москвичей было 9 мужчин. Некоторые из пассажиров были артистами, но ни одна из иногородних женщин артисткой не была. Всего иногородних мужчин было 18. Из них 13 — не были артистами. Среди пассажиров, не являвшихся артистами, были 16 мужчин и 11 женщин. 5 москвичей не были артистами.

Сколько всего артистов было в самолёте?

Решение.

Построим диаграмму: маленький круг — это артисты, большой — не артисты, снаружи — всего человек. Нанесём на диаграмму то, что нам известно:

1) 9 мужчин — москвичи;
2) область «иногородние женщины-артистки» закрашена, так как иногородних артисток нет;
3) 18 иногородних мужчин, из них 13 — не артисты; следовательно, артистов: 18 – 13 = 5;
4) так как среди пассажиров 16 мужчин — не артистов, а иногородних мужчин (не артистов) 13, то москвичей: 16 – 13 = 3;
5) тогда московских мужчин-артистов: 9 – 3 = 6;
6) так как среди москвичей 5 не артистов, и из них только трое мужчин, значит: 5 – 3 = 2 женщины;
7) а так как среди не артистов было всего 11 женщин, и 2 из них — москвички, то 11 – 2 = 9 иногородних;
8) используем, что всего пассажиров 42, и узнаем, сколько было артистов. Для этого от общего числа отнимем число не артистов: 42 – (2 + 9 + 3 + 13) = 15 человек.

2. Графы

Исторически сложилось так, что теория графов зародилась именно в ходе решения головоломок двести с лишним лет назад. Первая работа о графах, принадлежащая швейцарскому математику Леонарду Эйлеру, появилась в 1736 году. Эйлер начал свою работу о графах с рассмотрения одной головоломки — так называемой «задачи о кёнигсбергских мостах». Город Кёнигсберг расположен на берегах реки Прегель и двух островах. Различные части города были соединены семью мостами. По воскресеньям горожане совершали прогулки по городу. Вопрос заключался в том, можно ли совершить прогулку таким образом, чтобы, выйдя из дома, вернуться обратно, пройдя в точности один раз по каждому мосту?

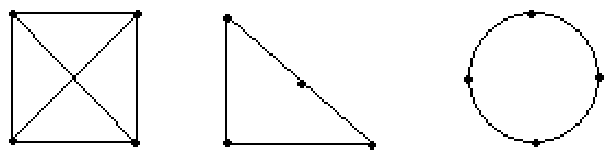
При необходимости провести взаимосвязь между достаточно большим количеством неких объектов мы, скорее всего, непроизвольно начнём рисовать на бумаге точки (кружочки), изображающие наши объекты, и соединять эти точки линиями или стрелками, отображающими интересующие нас отношения между рассматриваемыми объектами.

Имея перед глазами рисунок (схему), легче оценить представляющие интерес взаимозависимости между объектами. Такие схемы широко используют в различных областях: это и электрические цепи, и диаграммы, и сети коммуникаций, и схемы метрополитена, и схемы маршрутных автобусов, и т. д.

Итак, ***граф*** — это фигура, состоящая из точек (называемых вершинами) и отрезков, соединяющих некоторые из этих вершин. ***Вершина графа*** — либо конец какого-нибудь ребра графа, либо изолированная точка графа. ***Ребро графа*** — кривая, соединяющая две вершины графа и не содержащая других вершин.

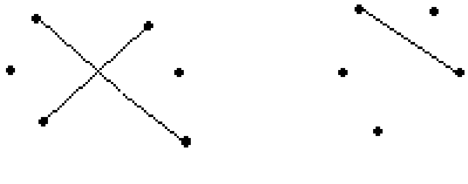
Замечания.

1. При изображении графов на рисунках или схемах отрезки могут быть прямолинейными или криволинейными (дугами). Длины отрезков и размещение точек произвольны. На данном рисунке изображён один и тот же граф:

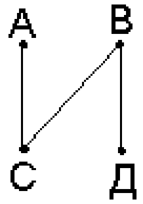


2. Вершины графа на рисунке выделяются обычно кружочками хотя бы потому, что не всегда точки пересечения рёбер принимаются за вершины графа.

Изолированными вершинами называются вершины, которые не соединены ни с какими другими вершинами графа.



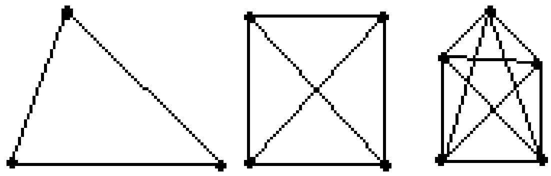
Степенью вершины графа называется число выходящих из него рёбер, при этом вершины называют чётными, если степень вершины чётна и нечётные, если степень вершины нечётна.



Пример:
степень А, Д = 1;
степень С, В = 2.

Граф называют ***простым***, если две вершины соединяет не более одного ребра. В противном случае граф называют ***мультиграфом***.

Граф называется ***полным***, если каждая его вершина соединена со всеми другими вершинами. Граф является полным, если каждые две различные его вершины соединены одним и только одним ребром. Примеры полных графов:



Замечания.

1. В полном графе каждая его вершина принадлежит одному и тому же числу рёбер.

2. Неполный граф можно преобразовать в полный с теми же вершинами, добавив недостающие рёбра.

Вершины в графе могут отличаться друг от друга тем, скольким рёбрам они принадлежат.

Задача 1. Лист бумаги Плюшкин разрезает на три части. Некоторые из полученных листов он также разрезает на три части. Несколько новых листов он вновь разрезает на три части и т. д.

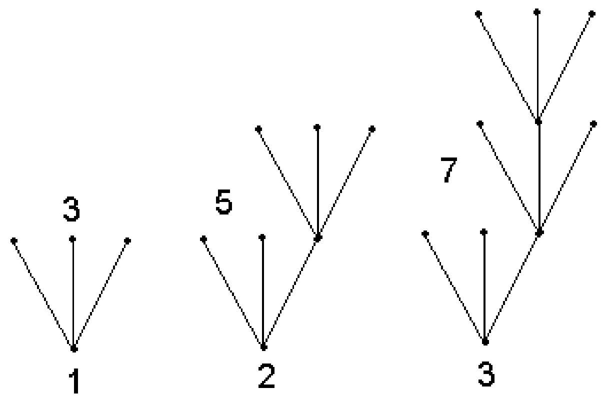
Сколько Плюшкин получает листов бумаги, если разрезает k листов?

Решение.

Удобно нарисовать схему: нижнее число показывает, сколько листов на данном шаге изрезано, а верхнее — сколько листов получилось после нарезки. Легко заметить, что число получившихся листов на 1 больше удвоенного числа порезанных:

3 = 2*1 + 1, 5 + 2*2 + 1, 7 = 2*3 + 1.

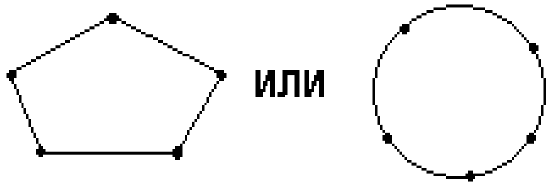
Значит, если разрезать k листов, получим $2k + 1$ лист.



Задача 2. Утверждают, что в одной компании из 5 человек каждый знаком с двумя и только с двумя другими. Возможна ли такая компания?

Решение.

Изобразим схему условия задачи:



Так как нам удалось изобразить условие, значит, такая компания действительно существует.

Теорема. *В графе сумма степеней всех его вершин — число, равное удвоенному числу рёбер графа.*

Задача 3. Девять шахматистов проводят турнир в один круг (каждый играет с каждым по одному разу).

Покажите, что в любой момент найдутся двое, закончившие одинаковое число партий.

1 способ (принцип Дирихле).

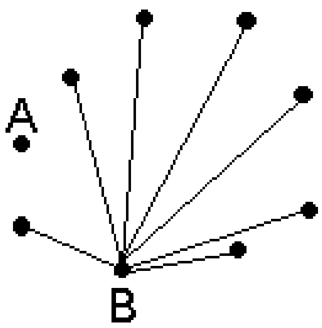
1) Хотя бы один не сыграл — тогда число возможных сыгранных партий: 0, 1, 2, 3, 4, 5, 6, 7 (всего 8), а шахматистов 9. Следовательно, хотя бы двое сыграли одинаковое число партий.

2) Все сыграли, по крайней мере, одну игру — тогда число игр: 1, 2,…, 8. Но шахматистов 9, значит, хотя бы два сыграли одинаковое число игр.

2 способ (с помощью графов).

Переведём условие на язык графов. Шахматисты — вершины графа. Ребром соединены шахматисты, которые сыграли между собой. Степень каждой вершины — число партий, сыгранных данным игроком.

Покажем, что во всяком графе с 9 вершинами существуют вершины одинаковой степени. Каждая вершина может иметь степень: 0,…, 8.



Пусть существует граф, вершины которого имеют разную степень: 0,…, 8. Но этого не может быть, так как если в графе вершина **A** имеет нулевую степень (изолирована), то в нём не найдётся вершины **B** со степенью 8. **B** должна быть соединена со всеми вершинами графа, а значит и с **A**. Получили противоречие. Следовательно, в самом начале мы сделали неправильное предположение, поэтому существуют две вершины одинаковой степени, то есть найдутся в любой момент два игрока, сыгравшие одинаковое число партий.

С помощью графов можно также решать и логические задачи.

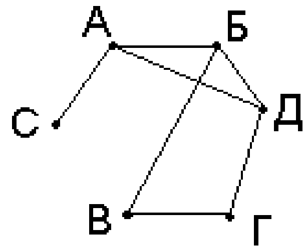
Задача 4. Однажды кто-то из учеников принёс учительнице цветы и поставил в вазу на столе. Когда собрались ребята, учительница спросила: «А знаете ли вы, кто принёс цветы?». Были высказаны различные предположения:

А и Б;
А и С;
А и Д;
Б и В;
Б и Д;
В и Г;
Г и Д.

Учительница сказала, что в одном из этих предположений одно имя названо правильно, а второе — нет. Во всех остальных — оба имени названы неверно.

Кто принёс цветы?

Решение.



Так как правильное имя названо только в одном предположении, то нужно искать вершину, принадлежащую одному ребру. А значит ученик С принёс цветы.

Путём (из вершины А в вершину В) в графе называется последовательная цепочка смежных рёбер (имеющих общую вершину), которая начинается в вершине А и заканчивается в вершине В. Путь может проходить через ребро только один раз.

Замечание. Запись пути в графе зависит от того, как определены его рёбра. Если рёбра определены с помощью вершин (в **задаче 4** рёбра: АС, АБ, БД и т. д.), то записывают последовательность тех вершин, через которые проходит путь. Если же рёбра имеют собственные названия, то выписывается последовательность из этих названий.

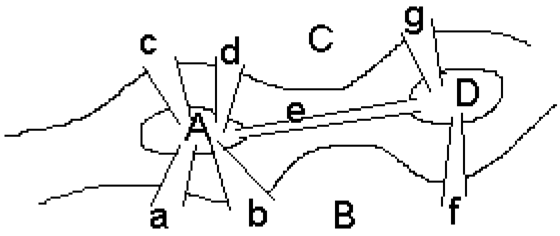
Пример. Путь из С в Г: САБДГ или САДГ.

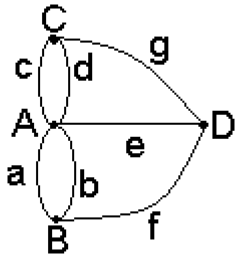
Граф называют ***связанным***, если любые две его вершины можно соединить хотя бы одним путём. В противном случае граф называют ***несвязанным.***

Вам, наверное, приходилось встречать задачи, в которых предлагается обвести ту или иную фигуру, не отрывая карандаш от бумаги. При этом запрещается проводить карандаш по одной линии несколько раз. Понятно, что аналогичное задание может быть дано относительно некоторого графа.

Задача 5 (задача о кёнигсбергских мостах). Бывший Кёнигсберг (ныне Калининград) расположен на реке Прегель (Преголи). В пределах города река омывает два острова. С берегов на острова были перекинуты мосты. Старые мосты не сохранились, но осталась карта города, где они изображены. Кёнигсбергцы предлагали приезжим следующую задачу: пройти по всем мостам и вернуться в начальный пункт, причём на каждом мосту следовало побывать только один раз.

Прогуляться по городским мостам предложили и Эйлеру. После безуспешной попытки совершить нужный обход он начертил упрощённую схему мостов. Получился граф, вершины которого — части города, разделённые рекой, а рёбра — мосты.





Прежде чем обосновать невозможность требуемого маршрута, Эйлер рассмотрел и другую, более сложную карту. В итоге он доказал общее утверждение: для того чтобы можно было обойти все рёбра графа по одному разу и вернуться в исходную вершину, необходимо и достаточно выполнение следующих двух условий:

1) из любой вершины графа должен существовать путь по его рёбрам в любую другую вершину (граф должен быть связанным);

2) из каждой вершины должно выходить чётное количество рёбер (степень каждой вершины чётная).

Замкнутый путь, проходящий по одному разу по всем рёбрам графа, называют (с тех пор) ***эйлеровым циклом***.

А как же дело обстоит с кёнигсбергскими мостами? Здесь, при каждой из четырёх вершин — нечётное число рёбер. Так что нет не только эйлерова цикла, но и пути из одной вершины в другую, проходящего по всем рёбрам графа.

Существует много других интересных задач, при решении которых используются графы.

3. Комбинаторные задачи

Человеку часто приходится иметь дело с задачами, в которых нужно подсчитать число всех возможных способов расположения некоторых предметов или число всех возможных способов осуществления некоторого действия. Сколькими способами можно расположить 50 человек в очереди в кассу за билетами в кино? Сколькими способами могут быть распределены золотая, серебряная и бронзовая медали на чемпионате Европы по футболу? Задачи такого типа называются ***комбинаторными***.

С комбинаторными вычислениями приходится иметь дело представителям многих специальностей: учёному-химику — при рассмотрении различных возможных типов связи атомов и молекул, биологу — при изучении различных возможных последовательностей чередования аминокислот в белковых соединениях, конструктору вычислительных машин, агроному, рассматривающему различные возможные способы посевов на нескольких участках, диспетчеру — при составлении графика движения. Комбинаторные соображения лежат в основе решения многих задач теории вероятностей.

Рассмотрим два основных комбинаторных принципа (правила), которые часто применяются при комбинаторных расчётах.

1. ***Правило произведения***: *если элемент x из множества X может быть выбран m способами, и после каждого такого выбора элемент y из множества Y может быть выбран n способами, то элементы x и y могут быть выбраны* $m \cdot n$ *способами.*

Мы сформулировали правило произведения для двух множеств, но оно может быть обобщено на любое конечное число множеств. Это значит, что если в задаче происходит выбор нескольких элементов из нескольких множеств, причём выбор сначала одного элемента, затем ещё одного, после этого следующего элемента и т. д., то для подсчёта числа выборов одновременно всех элементов, нужно применить принцип произведения.

Сформулируем второй комбинаторный принцип сразу в общем виде.

2. ***Правило суммы:*** *Если элемент* x_1 *из множества* X_1 *может быть выбран* m_1 *способами, элемент* x_2 *из множества* X_2 *может быть выбран* m_2 *способами, элемент* x_3 *из множества* X_3 *может быть выбран* m_3 *способами,…, элемент* x_k *из множества* X_k *может быть выбран* m_k *способами, причём любой выбор какого-то элемента не зависит от выбора остальных, то элемент* x_1 *или* x_2*, или* x_3*, или…, или* x_k *может быть выбран* $m_1 + m_2 + ... + m_k$ *способами.*

Таким образом, если происходит выбор одного элемента из нескольких, то для подсчёта числа выборов этого элемента можно использовать правило суммы.

Прежде чем перейти к определению комбинаторных соединений, введём понятия, которые нам потребуются.

В дальнейшем мы будем рассматривать только конечные множества, т. е. множества, состоящие из конечного числа элементов. В частности, если множество состоит из *n* элементов, то будем называть его для краткости ***n- множеством***. Некоторая совокупность элементов данного *n*- множества называется ***выборкой***. Число элементов в выборке называется ***длиной выборки***.

Если дано множество *X* — множество цифр, т. е. $X = \{0,1,2,3,4,5,6,7,8,9\}$, то говорят, что это множество состоит из 10 элементов. Любой набор цифр, состоящий, например, из 4 цифр, будет являться выборкой длины 4.

Замечание. Некоторую совокупность элементов из данного конечного множества можно выбирать по-разному:

1) – если при составлении выборки учитывается порядок следования элементов в выборке, тогда выборка называется ***упорядоченной***;
   – если же при составлении выборки не учитывается порядок следования элементов в ней, то выборка называется ***неупорядоченной***;

2) – если при составлении выборки в неё может быть включён один и тот же элемент множества несколько раз, то выборка называется ***выборкой с повторениями***;
– если же элементы множества входят в выборку только по одному разу, то выборка называется ***выборкой без повторений***.

Рассмотрим три основных вида комбинаторных соединений: размещения, перестановки и сочетания.

Размещением с повторениями из m элементов данного n-множества называется упорядоченная выборка с повторениями длины m из элементов данного множества.

Таким образом, если характер выборки: 1) упорядоченная, 2) с повторениями, то выборка является размещением с повторениями. Число размещений с повторениями из m элементов данного n-множества обозначается $\widetilde{A}_n^m$ и вычисляется по формуле: $\widetilde{A}_n^m = n^m$.

Размещением без повторений из m элементов данного n-множества называется упорядоченная выборка без повторений длины m из элементов данного множества.

Таким образом, если характер выборки: 1) упорядоченная, 2) без повторений, то выборка является размещением без повторений. Число размещений без повторений из m элементов данного n-множества обозначается A_n^m и вычисляется по формуле: $A_n^m = \dfrac{n!}{(n-m)!}$.

Замечание. Символ $n!$ читается как n – факториал и означает произведение первых n натуральных чисел: $n! = 1 \cdot 2 \cdot 3 \cdot ... \cdot n$.

Перестановкой без повторений из n элементов данного n-множества называется упорядоченная выборка без повторений длины n из элементов данного множества.

Таким образом, если характер выборки: 1) упорядоченная, 2) без повторений, и выбирают все элементы n-множества, то выборка является перестановкой без повторений. Число перестановок без повторений из n элементов обозначается P_n и вычисляется по формуле: $P_n = n!$

Сочетанием без повторений из m элементов данного n-множества называется любое подмножество из m элементов данного n-множества.

Таким образом, если характер выборки: 1) неупорядоченная, 2) без повторений, то выборка является сочетанием без повторений. Число сочетаний без повторений из m элементов данного n-множества обозначается C_n^m и вычисляется по формуле: $C_n^m = \dfrac{n!}{m!(n-m)!}$.

Из приведённых определений видно, что классификация комбинаторных соединений ведётся по характеру выборки, поэтому при решении конкретной задачи очень важно определить характер выборки.

Если в комбинаторной задаче дано одно множество, из которого происходит выбор, и на выборку (кроме характера) не накладываются дополнительные условия, то такую задачу будем называть ***простой комбинаторной задачей*** и решать по алгоритму:

1) Определить множество, из которого производится выборка, и число элементов n в этом множестве.
2) Определить выборку и число элементов m в выборке.
3) Определить характер выборки.
4) По характеру выборки определить, с каким комбинаторным соединением мы имеем дело, и выбрать формулу для вычисления.
5) По выбранной формуле произвести вычисления.

Если в задаче определяется несколько множеств, или на выборку (кроме характера) накладываются дополнительные условия, то такую задачу будем называть ***сложной комбинаторной задачей*** и решать по следующему алгоритму:

1) По условию задачи определить, какую задачу мы имеем: сложную или простую.
2) Разбить сложную комбинаторную задачу на несколько простых задач.
3) Решить каждую простую задачу по приведённому выше алгоритму.
4) Определить, какой комбинаторный принцип нужно применить в этой задаче:
   – если выбираем несколько элементов по принципу «сначала… потом…», то необходимо применить правило произведения;
   – если выбираем один элемент из нескольких по принципу «или первый, или второй, или …», то применяем правило суммы.
5) В соответствии с выбранным принципом проводим вычисления.

Задачи для самостоятельного решения

1. В кругу сидят четыре котёнка: Барсик, Дымок, Васька и Тимофей. Их цвета: белый, серый, рыжий и чёрный.

– Барсик не рыжий;
– Дымок сидит между белым и чёрным котятами;
– между Дымком и рыжим котёнком сидит Тимофей;
– Тимофей не белый.

Определите цвет каждого котёнка.

2. На столе лежат в ряд четыре фигуры: треугольник, круг, прямоугольник и ромб. Они окрашены в разные цвета: красный, синий, жёлтый и зелёный. Известно, что:

– красная фигура лежит между зелёной и синей;
– справа от жёлтой фигуры лежит ромб;
– круг лежит правее треугольника и ромба;
– треугольник не лежит с краю;
– синяя и жёлтая фигуры не лежат рядом.

Определите цвет и порядок расположения фигур.

3. Три подруги — Надя, Валя и Маша — вышли гулять в белом, красном и чёрном платьях. Туфли их были тех же трёх цветов, но только:
– у Нади цвета туфель и платья совпадали;
– при этом у Вали ни платье, ни туфли не были чёрными;
– Маша была в красных туфлях.

Определите цвета платьев и туфель каждой из них.

4. Четверо друзей — Дима, Миша, Лёня и Максим — получили в подарок по котёнку. Их назвали Пушок, Елисей, Фантик и Мурлыка. Каждый из ребят выбрал себе котёнка любимого цвета. На расспросы они ответили следующее:
– Фантик не рыжий, а Мурлыка не серый.
– Пушок не белый, а Елисей не серый.
– Миша выбрал чёрного котёнка, а Максим — Мурлыку.
– Лёня взял Елисея, а Дима — белого котёнка.
– Пушок не серый, и Дима не взял Фантика.

Какого цвета котёнок у каждого из друзей, если во всех высказываниях одно утверждение истинно, а другое нет?

5. В небольшой деревушке жили лесорубы, охотники и рыбаки. Лесорубы всегда говорили правду. Охотники, если им задавали несколько вопросов, отвечали поочерёдно то правду, то неправду (в ответ на первый вопрос они могут сказать и неправду). Рыбаки всегда говорили неправду. За круглым столом сидели трое жителей деревни в таком порядке (по часовой стрелке): Боб, Дик и Пит. Приезжий задал всем по очереди три вопроса: 1. Кто Ваш сосед слева? 2. Кто Ваш сосед справа? 3. Кто Вы?

Вот как они ответили на эти вопросы:

Боб: 1. Рыбак. 2. Охотник. 3. Лесоруб.

Дик: 1.Охотник. 2. Охотник. 3. Лесоруб.

Пит: 1. Рыбак. 2. Лесоруб. 3. Лесоруб.

Кем на самом деле были Боб, Дик и Пит?

6. Журналист прибыл в аэропорт, чтобы побеседовать с Фёдоровым, Григорьевым и Даниловым — лётчиком, бортинженером и штурманом одного самолёта. Знакомый пилот сообщил ему следующие факты:
– Данилов не лётчик;
– Фёдоров не бортинженер;
– Данилов не бортинженер;

– Григорьев не штурман.

Когда журналист начал беседовать с экипажем, выяснилось, что из всех четырёх фактов соответствует действительности только один.

Какая специальность у каждого члена экипажа?

7. В соревнованиях по плаванию три пловца: Михаил, Андрей и Валерий — пришли к финишу почти одновременно, и между болельщиками разгорелся спор:

– один утверждал, что Михаил — второй, а Валерий — третий;

– другой доказывал, что Михаил — третий, а Валерий — первый;

– третий говорил, что Андрей — второй, а Валерий — первый.

Когда судьи объявили результаты заплыва, оказалось, что каждый болельщик был прав только наполовину.

Как распределились места?

8. В одной школе учатся три друга: Сергей, Коля и Максим. Их фамилии: Петров, Семёнов и Иванов.

– Сергей учится в 5 классе;

– мама Коли — инженер;

– Иванов учится в 6 классе, его мама — бухгалтер;

– Сергей и Семёнов болеют за разные футбольные клубы.

У кого из друзей какая фамилия?

9. Четыре ученика (Андрей, Борис, Владимир и Геннадий) заняли первые четыре места на районной математической олимпиаде, причём двое из них не делили между собой какие-нибудь два места. На вопрос: «Какое место занял каждый из них?» участники дали три разных ответа:

1) Андрей — первое, Борис — второе;
2) Андрей — второе, Геннадий — третье;
3) Владимир — второе, Геннадий — четвёртое.

Причём в каждом из ответов одна часть истинна, а другая ложна.

Какое место занял каждый из участников олимпиады?

10. После зимних каникул классный руководитель спросил, посещали ли ученики кино, театр, цирк на каникулах. Оказалось, что из 36 учеников:

– только двое не были ни в кино, ни в театре, ни в цирке;

– в кино побывали 25 человек, в театре — 11, в цирке — 17;

– в кино и театре — 6, в кино и цирке — 10, в театре и цирке — 4.

Сколько человек побывали и в кино, и в театре, и в цирке?

11. Мама купила яблоки для троих своих детей. Дети должны поделить эти яблоки поровну между собой. Первым пришёл из школы старший сын, взял себе третью часть и ушёл гулять. Второй пришла сестра и, думая, что она пришла первой, взяла себе третью часть, и ушла. Последним пришёл младший брат и взял третью часть остатка. После этого осталось 8 яблок.

Сколько яблок было всего? Как дети должны поделить оставшиеся яблоки?

12. Дорога от дома до хозяйственного магазина занимает у Пети 40 мин. Однажды по дороге в магазин он вспомнил, что забыл дома список покупок. Если он пойдёт дальше, то придёт за 15 мин до обеденного перерыва. Если вернётся обратно, то опоздает на 5 мин.

Какую часть пути успел пройти Петя?

13. Дама сдавала в багаж рюкзак, чемодан, саквояж:

– чемодан и рюкзак вместе весят 12 кг;

– чемодан и саквояж — 15 кг;

– саквояж и рюкзак — 9 кг.

Сколько весит каждый предмет в отдельности?

14. Однажды чёрт предложил бездельнику заработать. «Как только ты перейдёшь через этот мост, — сказал он, — твои деньги удвоятся. Можешь переходить по нему сколько хочешь раз, но после каждого перехода отдавай мне за это 24 рубля». Бездельник согласился и … после третьего перехода остался без денег.

Сколько денег у него было сначала?

15. Группа туристов отправилась в поход. В первый день они прошли 1/3 пути, во второй — 1/3 остатка, в третий — 1/3 нового остатка. В результате им осталось пройти 32 км.

Сколько километров был маршрут туристов?

16. Трое мальчиков имеют по некоторому количеству яблок. Первый мальчик даёт другим столько яблок, сколько каждый из них имеет. Затем второй мальчик даёт двум другим столько яблок, сколько каждый из них теперь имеет. В свою очередь и третий даёт каждому из двух других столько, сколько есть у каждого в этот момент. После этого у каждого из мальчиков оказывается по 8 яблок.

Сколько яблок было у каждого мальчика вначале?

17. Отец решил отдать сына на обучение и спросил учителя: «Скажите, сколько учеников у Вас в классе?» Учитель ответил: «Если придёт ещё столько же учеников, сколько имею, и полстолько, и четвертая часть, и твой сын, тогда будет у меня сто учеников».

Сколько же учеников было в классе?

18. Мальчик прочитал книгу за 3 дня:

– в первый день он прочитал 0,2 всей книги и ещё 16 страниц;

– во второй день — 0,3 остатка и ещё 20 страниц;

– в третий — 0,75 нового остатка и последние 30 страниц.

Сколько страниц в книге?

19. В стране 8 городов, названия которых начинаются на буквы: А, Б, В, Г, Д, Е, Ж, З, и 8 непересекающихся дорог между городами: А и Б, Е и Д, Б и Ж, З и А, В и Г, Г и Д, Ж и З, В и Е.

Можно ли по этим дорогам проехать из А в Г?

20. В квартирах № 1, 2, 3 жили три друга: Айдар, Тима и Саша. Известно, что:

– в квартирах № 1 и № 2 жил не Айдар;

– Тима жил не в квартире № 1.

В какой квартире жил каждый из друзей?

21. Три друга — Алёша, Боря и Вова — после школы едут домой на различном транспорте: автобусе, маршрутке, трамвае. Однажды после уроков Алёша пошёл проводить своего друга до остановки автобуса. Когда мимо них проходила маршрутка, третий друг крикнул из окна: «Боря, ты забыл в школе тетрадку!»

Кто на чём ездит домой?

22. Три подруги — Тамара, Валя и Лида — были в белом, красном и голубом платьях. Их туфли были тех же цветов. Только:

– у Тамары цвета платья и туфель совпадали;

– Валя была в белых туфлях;

– ни платье, ни туфли Лиды не были красными.

Определите цвет платья и цвет туфель каждой из подруг.

23. Три друга — Лёша, Сергей и Денис — купили щенков разной породы (щенка таксы, щенка колли и щенка овчарки) и дали им клички. Известно, что:

– щенок Лёши темнее по окрасу, чем такса, Леси и Гриф;

– щенок Сергея старше Грифа, овчарки и таксы;

– Джек и такса всегда гуляют вместе.

У кого какой породы щенок? Назовите клички щенков.

24. В столовой на горячее можно заказать щуку, грибы и баранину, на гарнир — картофель и рис, а из напитков — чай и кофе.

Сколько различных вариантов обедов можно составить из указанных блюд?

25. Алла решила маме на День рождения подарить букет цветов (розы, тюльпаны или гвоздики) и поставить их или в вазу, или в кувшин.

Сколькими способами это можно сделать?

26. Сколько трёхзначных чисел можно составить из цифр: 1, 3, 5, 7, используя в записи числа каждую из них не более одного раза?

27. В первенстве класса по шашкам 5 участников: Аня, Боря, Влад, Гриша, Даша. Первенство проводится по круговой системе — каждый из участников играет с каждым из остальных один раз. К настоящему времени некоторые игры уже проведены:

– Аня сыграла с Борей, Владом и Дашей;
– Боря сыграл с Аней и Гришей;
– Влад — с Аней и Дашей;
– Гриша — с Борей;
– Даша — с Аней и Гришей.

Сколько игр проведено к настоящему времени и сколько ещё осталось?

28. Сколько всех трёхзначных чисел можно составить из нечётных цифр?

29. Сколько различных наборов по 3 книги можно составить из 10-томного собрания писателя A, 6-томного — писателя B и 12-томного — писателя C, если в наборе должно быть по одной книге каждого писателя?

30. Сколько различных списков участников соревнования можно составить, если в соревновании принимают участие 6 учеников?

ИНФОРМАТИКА

Решение олимпиадных задач по программированию

Е.А. Редько,
старший преподаватель кафедры математики и информационных технологий ФГБОУ ВО ПИ ТОГУ

1. Обработка строк.
Строка как массив символов, понятие строки, описание, инициализация

Строковый тип — ещё один представитель структурированных типов. Несмотря на то, что строку определяют как массив символов, алгоритмы обработки строк имеют свои особенности. Кроме того, во многих языках программирования для реализации алгоритмов обработки строк существует набор готовых функций. В языке *C* эти функции становится возможным использовать при подключении заголовочного файла string.h.

Итак, строка — это массив символов:

char	stroka	[25];
Базовый тип элементов	Имя строковой переменной	Длина строки

Последним символом строки обязательно должен быть нуль-символ (***нуль-терминатор***) с кодом 0 (добавляется автоматически в конце текстового значения). В тексте программы нуль-символ изображается: \0.

Если при объявлении строки не указана её длина, но выполнена одновременная инициализация, то длина строки равна количеству символов плюс нулевой символ:

```
char st1[]="Programming"; // длина st1 равна 11+1 .
```

Это необходимо помнить при объявлении с одновременной инициализацией и задавать длину строки на один символ больше. В противном случае возникнет синтаксическая ошибка:

```
char st1[11]="Programming"; // недопустимо .
```

Примечание: строка может быть инициализирована списком символов, как и массив, но в этом случае программист должен самостоятельно позаботиться о наличии последнего нуль-символа:

```
char st2[]={'H', 'e', 'l', 'l', 'o', '!', '\0'};
```

Ввод значения строковой переменной с клавиатуры

Рассмотрим функции библиотеки стандартного ввода/вывода, позволяющие получить значение строковой переменной с клавиатуры.

1 способ. Функция scanf() со спецификатором %s:

scanf("%s", stroka);

Такой вариант позволит считывать текст до первого разделителя, которым может быть не только символ перехода на новую строку, но и символ табуляции или пробел. Поэтому, если пользователь вводит текст, состоящий из нескольких слов, разделённых пробелами, то считано в строковую переменную будет только первое слово.

2 способ. Функция gets():

gets(stroka);

Данная функция предназначена именно для считывания строк.

3 способ. Функция scanf() со спецификатором %[^\n]:

scanf("%[^\n]", stroka);

Инструкция %[^\n] означает, что считываться должны все символы, за исключением символа \n перехода к новой строке.

Вывод строки на экран

Вывод строки может быть реализован одной из функций:

puts(stroka); // вариант 1

printf("%s", stroka); // вариант 2

Таблица 1

Основные функции для обработки строк

Функция	Тип возвращаемого результата	Назначение
strcpy(st1, st2)		копирует содержимое строки st2, включая нулевой символ, в строку st1 (copy)
strcat(st1, st2)		добавляет справа к строке st1 содержимое строки st2 (concatenation)
strcmp(st1, st2)	int	сравнивает содержимое строк st2 и st1 (compare): если st1 < st2, то результат = –1; если st1 = st2, то результат = нулю;

		если st1 > st2, то результат равен 1
strstr(st1, st2)	char *	возвращает указатель на первое появление подстроки st2 в строке st1
strchr(st, sh)	char *	возвращает указатель на первое появление символа ch в строке st
strlen(st)	int	возвращает длину строки st (length)
strrev(st)		изменяет порядок следования символов в строке на противоположный (reverse)
strdup(st)	char *	дублирует строку st (duplicate)
strlwr(st)		конвертирует символы строки st к нижнему регистру (low registrer)
strupr(st)		конвертирует символы строки st к верхнему регистру (up register)
atoi(st)	int	преобразует строку st в число целого типа (int)
atof(st)	double	преобразует строку st в число действительного типа
itoa(a, st, base)		преобразует число целого типа a в строку st (base — основание системы счисления)
gcvt(a, dec, st)		преобразует число действительного типа a в строку st. Значение dec указывает на число десятичных разрядов (не более 18)

Примеры решения задач для обработки строк

Пример 1.

Программа иллюстрирует (рис. 1) результат использования функций считывания (gets()), дублирования (strdup()), конвертации к верхнему регистру (strupr()), определения длины строки (strlen()), изменения порядка следования символов (strrev()) и конкатенации (strcat()).

Листинг программы

```
#include <stdio.h>
#include <string.h>
```

```
int main()
{
   char st1[12],*s;
   gets(st1);
   s=strdup(st1);
   strupr(s);
                          printf("%s \n string length = %d \n", st1, strlen(st1));
   printf("Up register: \t%s \n",s);
   strrev(s);
   printf("Reverse: \t%s \n",s);
                                      char st2[]={'H', 'e', 'l', 'l', 'o', ',', ' ', '\0'};
   strcat(st2, st1);
   printf("Concatenation: \t%s \n",st2);
   return 0;
}
```

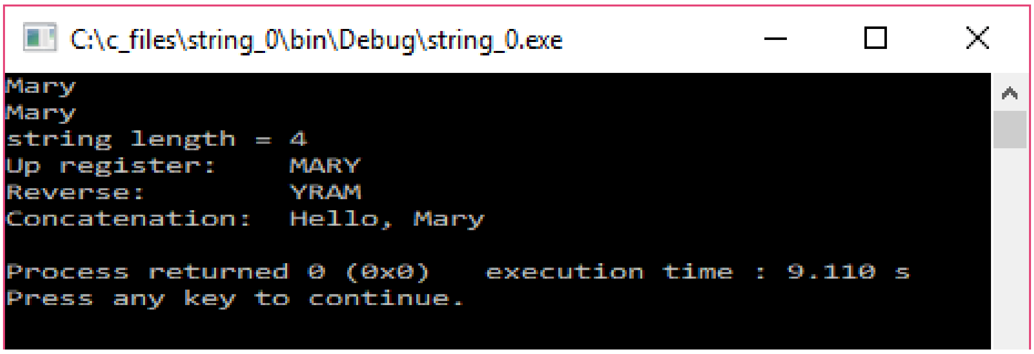


Рис. 1. Результат работы программы примера 1.

Пример 2. Задача «Слова».

Выделить и вывести на экран все слова произвольной строки. Слова отделяются друг от друга одним или несколькими пробелами (рис. 2).

Листинг программы

```
#include <stdio.h>
#include <string.h>

int main()
{
   char st[100], sl[100];
   int k=0, i;
   gets(st);
   strcat(st," ");
   int n=strlen(st);
   if (n<2) return 1;
   sl[0]='\0';
   for (i=0; i<n; i++)
      if (st[i] != ' ')
```

```
        {
           sl[k]=st[i];
           sl[k+1]='\0';
           k++;
        }
        else
        {
           if (strlen(sl)>0) puts(sl);
           sl[0]='\0';
           k=0;
        }
return 0;
}
```

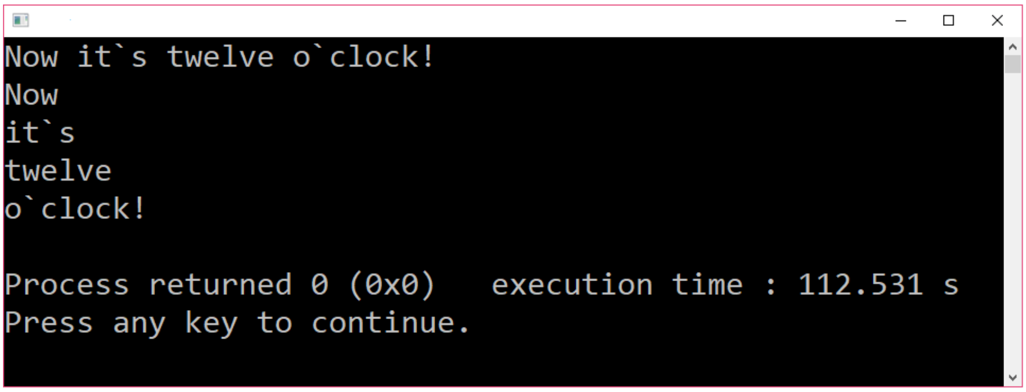


Рис. 2. Результат работы программы «Слова».

Пример 3. Задача «Клавиатура».

Для данной буквы английского алфавита нужно вывести справа стоящую букву на стандартной клавиатуре. При этом клавиатура замкнута, т.е. справа от буквы «p» стоит буква «a», от буквы «l» стоит буква «z», а от буквы «m» — буква «q».

Входные данные

Первая строка входного файла INPUT.TXT содержит один символ — маленькую букву английского алфавита.

Выходные данные

В выходной файл OUTPUT.TXT следует вывести букву, стоящую справа от заданной буквы, с учётом замкнутости клавиатуры.

Листинг программы

```
#include <iostream>
using namespace std;
int main()
{
   char klava[]="qwertyuiopasdfghjklzxcvbnmq";
   char symb;
   cin>>symb;
   int i=0;
```

```
  while(i<26&&symb!=klava[i])i++;
  cout<<klava[i+1];
  return 0;
}
```

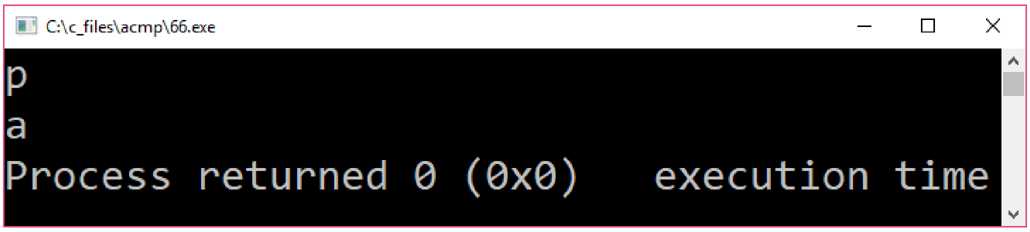


Рисунок 3. Пример работы программы к задаче «Клавиатура»

Пример 4. Задача «Нули».

Требуется найти самую длинную непрерывную цепочку нулей в последовательности нулей и единиц.

Входные данные

В единственной строке входного файла INPUT.TXT записана последовательность нулей и единиц (без пробелов). Суммарное количество цифр от 1 до 100.

Выходные данные

В единственную строку выходного файла OUTPUT.TXT нужно вывести искомую длину цепочки нулей.

Листинг программы

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
 char p[102];
 cin>>p;
 int max=0;
 int count=0;
 int i=0;
 while(i<strlen(p))
 {
  if(p[i]=='0')count++;
  else {
   if (count>max)max=count;
        count=0;
  }
  i++;
 }
 if (count>max)max=count;
 cout<<max;
 return 0;
```

```
}
```

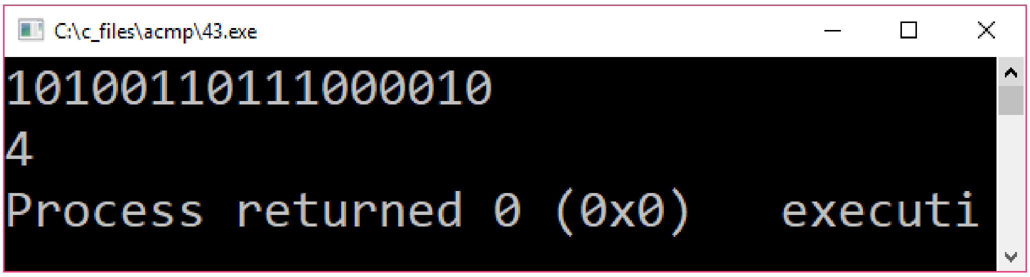


Рисунок 4. Пример работы программы к задаче «Нули»

Пример 5. Задача «Число E».

Выведите в выходной файл округлённое до n знаков после десятичной точки число E. В данной задаче будем считать, что число Е в точности равно 2.7182818284590452353602875.

Входные данные

Входной файл INPUT.TXT содержит целое число n ($0 \leq n \leq 25$).

Выходные данные

В выходной файл OUTPUT.TXT выведите ответ на задачу.

Листинг программы

```
#include <iostream>
using namespace std;
int main()
{
 char e[]="2.7182818284590452353602875";
 int i,n;
 cin>>n;
 if(n==0)cout<<3;
 else {
  if(e[n+2]-'0'>=5)
  {
   e[n+1]+=1;
   e[n+2]='\0';
  }
  else e[n+2]='\0';
  cout<<e;
 }
 return 0;
}
```

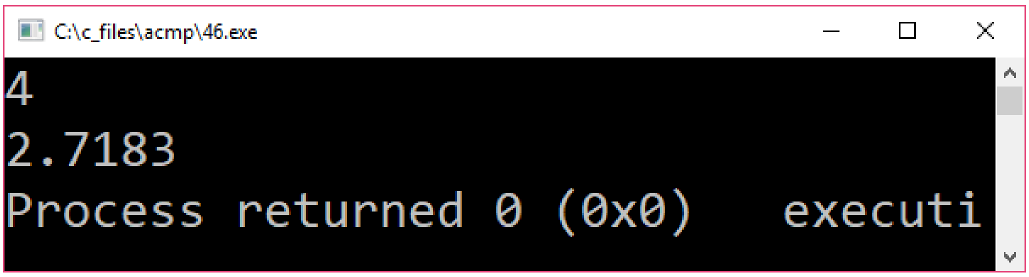


Рисунок 5. Пример работы программы к задаче «Число E»

2. Комбинаторные объекты. Рекурсивные функции

В современной теории алгоритмов принято разделять расчётные процедуры и функции на рекурсивные и итеративные. ***Рекурсивными подпрограммами*** (или просто ***рекурсией)*** принято называть такие процедуры или функции, которые в процессе выполнения вызывают сами себя.

Рекурсивные алгоритмы, в частности, могут использоваться для получения математических объектов, имеющих рекуррентное определение. Такими объектами являются, например, факториал или последовательность чисел Фибоначчи.

Рассмотрим рекурсивный алгоритм на языке C/C++ на примере вычисления n-го числа Фибоначчи:

```
int fib(int n)
{
    if (n==1 || n==2)
        return 1;
    else
        return fib(n-1)+fib(n-2);
}
```

Рекурсивная функция fib( ) повторяет рекуррентное определение чисел последовательности Фибоначчи:

$$\mathrm{F_n} = \begin{cases} 1, \text{при } \mathrm{n} = 1 \text{ и } \mathrm{n} = 2 \\ \quad F_{\mathrm{n}-1} + F_{\mathrm{n}-2} \end{cases}$$

В теле описания функции fib( ) в одной из «веток» условного оператора мы видим обращение к самой же функции fib( ), но с другими параметрами: $\mathrm{n} - 1$ и $\mathrm{n} - 2$. Кроме того, при обращении к функции fib( ) с параметром $\mathrm{n} = 1$ или $\mathrm{n} = 2$ рекурсивного вызова не произойдёт, так как функция вернёт константное значение, равное единице.

Обобщая вышесказанное, можно сформулировать ***правила описания рекурсивной функции:***

1. Тело рекурсивной функции должно содержать решение самого простого варианта задачи (реализуется проверкой достижения параметром некоторого значения).

2. Рекурсивные вызовы функции должны сводить общие случаи решения задачи к более простым случаям (реализуется за счёт изменения параметра при рекурсивном вызове).

С рекурсивной функцией связано понятие ***глубины рекурсии*** — максимального числа рекурсивных вызовов функции без возвратов, которое происходит во время выполнения программы.

Построение дерева рекурсивных вызовов позволяет определить глубину рекурсии. Так, например, дерево рекурсивных вызовов функции, определяющей *n-е* число Фибоначчи (рис. 6), наглядно демонстрирует глубину «спуска» рекурсии при обращении к функции fib( ) с параметром 5. Так как высота дерева равна пяти, то и глубина рекурсивной функции fib( ) равна 5.

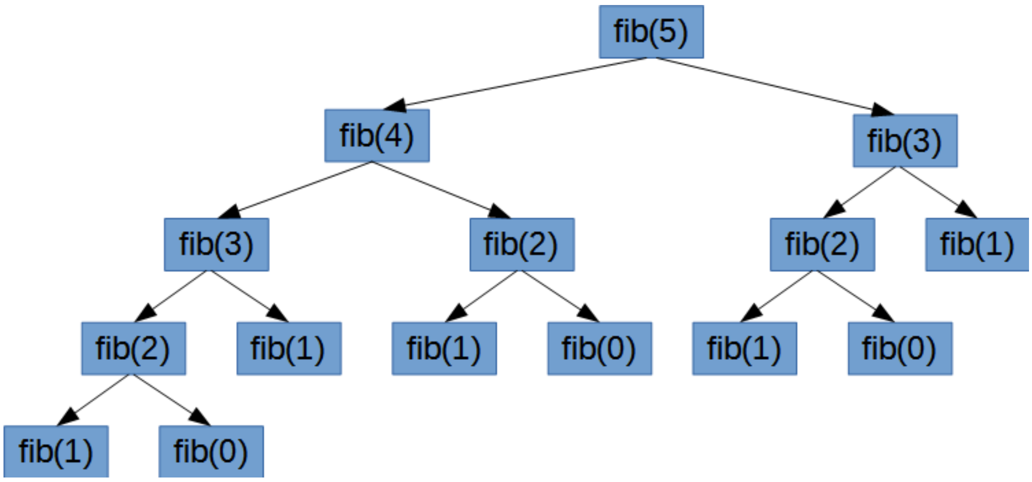


Рисунок 6. Дерево рекурсивных вызовов функции fib(5).

Примечание: рекурсивный алгоритм вычисления чисел Фибоначчи имеет экспоненциальную сложность как по времени, так и по используемой стековой памяти; на практике применяется только для небольших значений n.

Задачи для самостоятельного решения

1. Дано натуральное число n. Выведите все числа от 1 до n.
2. Дано натуральное число n. Выведите все числа от n до 1.
3. Дано натуральное число n. Выведите все его цифры, начиная с младшей.
4. Дано натуральное число n. Выведите все его цифры, начиная со старшей.

Класс С++, реализующий работу с массивом (vector). Шаблон описания вектора в С++

Класс *<vector>* является альтернативой применения встроенных массивов.

Для описания векторов в программе необходимо подключить библиотеку *<vector>*. Тогда можно выполнить один из вариантов описания вектора:

а) vector <базовый_тип> имя(размерность);

б) vector <базовый_тип> имя(размерность, значение по умолчанию);

Примеры описания векторов:

- ✓ *vector <int> A(10);* //массив из 10 целых чисел
- ✓ *vector <double> B(50, 0);* //массив из 50 вещественных чисел, инициализированных нулём
- ✓ *vector <char> list;* // пустой список символов
- ✓ *vector <string> text(15);* //массив из 15 строк

Некоторые методы класса <vector>

Таблица 2.

Перечень некоторых методов, реализованных в библиотеке <vector>

Метод	Описание
push_back(x)	Добавление элемента в конец массива. Параметром функции является значение, размещаемое в массиве. Тип этого параметра должен совпадать с базовым типом элементов массива
pop_back ()	Удаление последнего элемента массива. Нет параметров
front()	Вывод значения, расположенного в начале списка
back()	Вывод значения, расположенного в хвосте списка
insert(p,x)	Вставка элемента x в некоторое место массива, задаваемое указателем p (через итератор)

Примеры построения комбинаторных объектов

Перейдём к рассмотрению соединений — понятий комбинаторики, а именно: перестановок (P, permutations, фр.) и размещений (A, arrangements, фр.) [3].

Перестановкой без повторений из n элементов называется кортеж длины n, составленный из этих элементов.

Пример.

Пусть множество состоит из пяти элементов $\{1, 2, 3, 4, 5\}$. Тогда перестановками будут следующие кортежи:

12345
12354
12435
12453
12534
12543
и т.д.

Размещением из n элементов по m элементов (m≤n) называется упорядоченная выборка m элементов из данного множества n элементов.

Пример.

Пусть множество состоит из пяти элементов $\{1, 2, 3, 4, 5\}$. Тогда размещениями по 2 элемента будут следующие выборки:

12, 13, 14, 15, 21, 23, 24, 25, 31, 32, 34, 35, 41 и т.д.

Упорядоченные выборки m элементов с повторениями, которые составлены из основного множества n элементов, называются ***размещениями с повторениями из n элементов по m элементов***.

Пример.

Пусть множество состоит из пяти элементов $\{1\ , 2, 3, 4, 5\}$. Тогда размещениями с повторениями по 2 элемента будут следующие выборки: 11, 12, 13, 14, 15, 21, 22, 23, 24, 25, 31 и т.д.

Пример 1. Двоичные числа длины n.

```
#include <iostream>
#include <vector>

using namespace std;
void print_bin(vector<int> prefix,int n);
int main()
{
   int N=4;
   vector<int> bin_num(0);
   print_bin(bin_num,N);
   return 0;
}

void print_bin(vector<int> prefix,int n)
{
   static int k=0;
   if (n==0)
   {
      cout<<"binary number: ";
      for (int i=0;i<prefix.size();i++)
         cout<<prefix[i]<<" ";
      cout<<" count of "<<k<<endl;
      k++;
   }
   else
   {
      prefix.push_back(0);
      print_bin(prefix,n-1);
      prefix.pop_back();
      prefix.push_back(1);
      print_bin(prefix,n-1);
      prefix.pop_back();
   }
}
```

Результат работы программы:

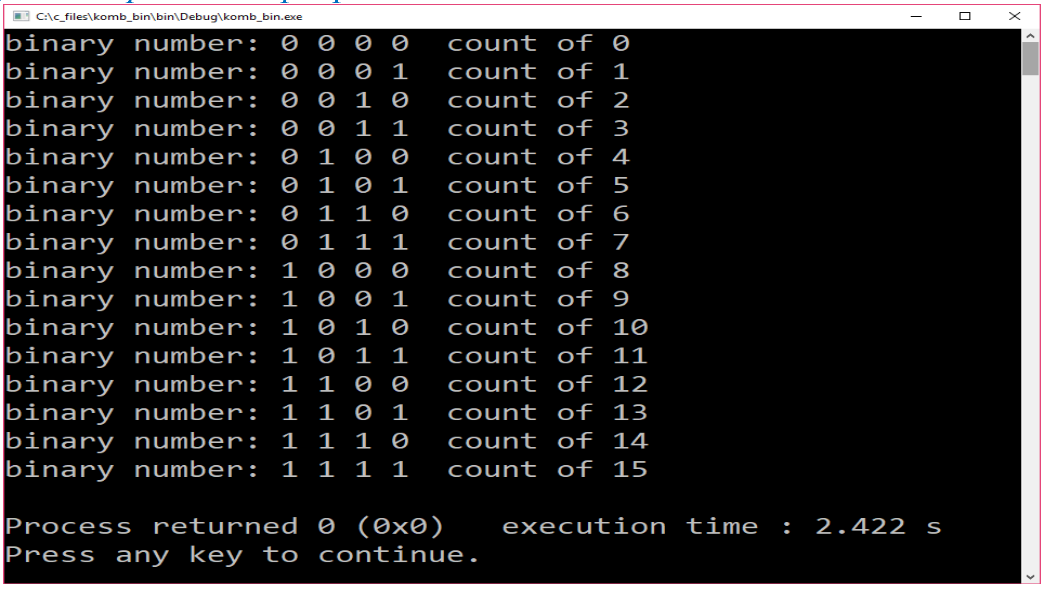


Рисунок 7. Создание двоичных чисел длины 4

Пример 2. Построение чисел длины n из m цифр.

```
#include <iostream>
#include <vector>

using namespace std;
void print_num(vector<int> prefix,int n,int m);
int main()
{
   int N=10,M=2;
   vector<int> num(0);
   print_num(num,N,M);
   return 0;
}

void print_num(vector<int> prefix,int n,int m)
{
   if (m==0)
   {
      for (int i=0;i<prefix.size();i++)
         cout<<prefix[i]<<" ";
      cout<<", ";
   }
   else
   {
      for (int k=0;k<n;k++)
      {
         prefix.insert(prefix.end(),k);
         print_num(prefix,n,m-1);
```

```
            prefix.pop_back();
        }
    }
}
```
Результат работы программы:

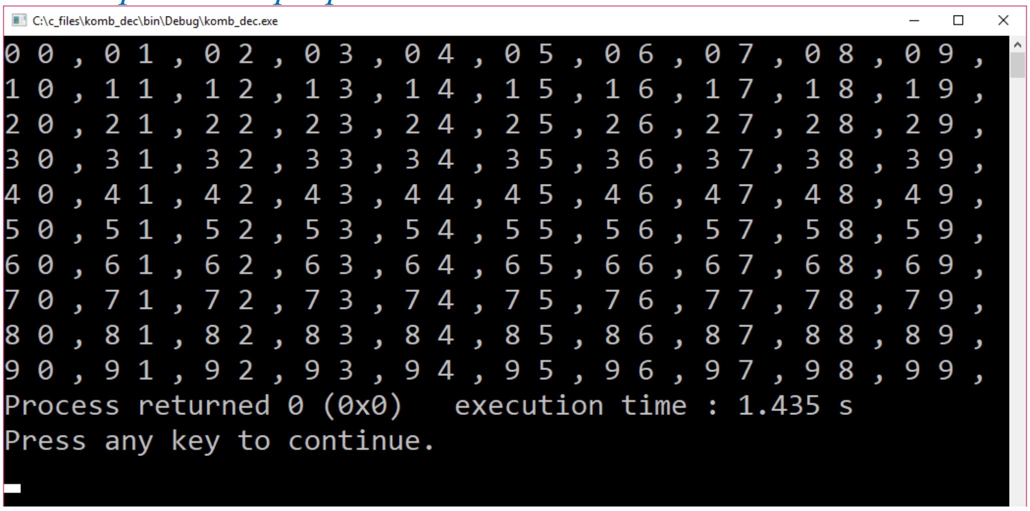


Рисунок 8. Формирование десятичных чисел длины 2

Пример 3. Перестановки без повторений.
```
#include <iostream>
#include <vector>

using namespace std;
bool search_digit(vector<int> prefix,int d);
void print(vector<int> prefix,int n,int m);
int main()
{
    int N=4,M;
    vector<int> permitation(0);
    print(permitation,N,M=N);
    return 0;
}

void print(vector<int> prefix,int n,int m)
{
    if (m==0)
    {
        for (int i=0;i<prefix.size();i++)
            cout<<prefix[i]<<" ";
        cout<<endl;
    }
    else
    {
        for (int k=0;k<n;k++)
```

```
        if (!search_digit(prefix,k))
        {
            prefix.insert(prefix.end(),k);
            print(prefix,n,m-1);
            prefix.pop_back();
        }
    }
}

bool search_digit(vector<int> prefix,int d)
{
    bool flag=false;
    for (int j=0;j<prefix.size();j++)
        if (prefix[j]==d) flag=true;
    return flag;
}
```

Результат работы программы:

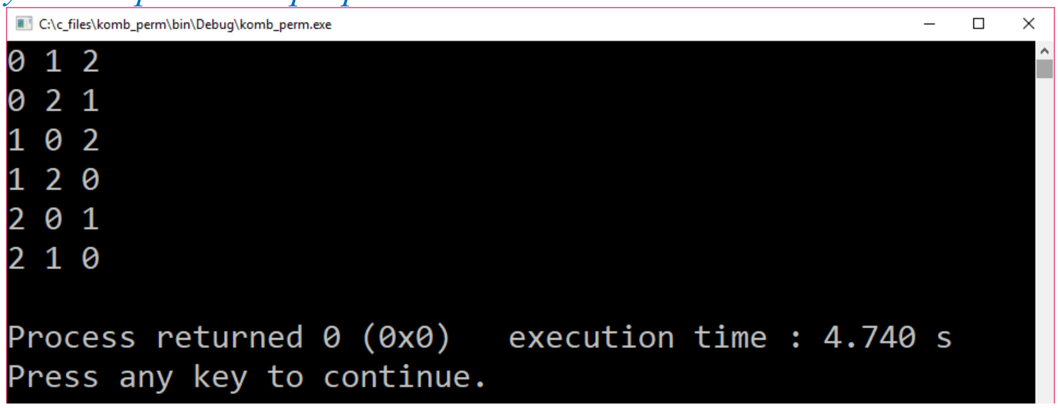


Рисунок 9. Генерация перестановок без повторений длины 3

Пример 4. Разбиения.

```
#include <iostream>
#include <vector>

using namespace std;
bool search_digit(vector<int> prefix,int d);
void print(vector<int> prefix,int n,int m);
int main()
{
    int N=4,M;
    vector<int> split(0);
    print(split,N,M=N);
    return 0;
}

void print(vector<int> prefix,int n,int m)
```

```
{
   if (m==0)
   {
      for (int i=0;i<prefix.size();i++)
         cout<<prefix[i]<<" ";
      cout<<endl;
   }
   else
   {
      for (int k=1;m-k>=0;k++)
      {
         prefix.insert(prefix.end(),k);
         print(prefix,n,m-k);
         prefix.pop_back();
      }
   }
}
```

Результат работы программы:

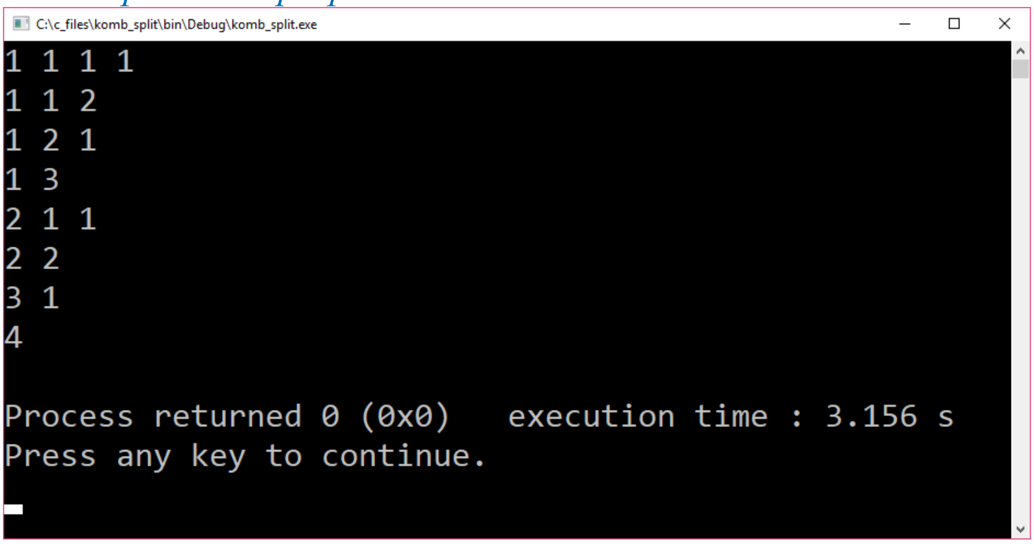


Рисунок 10. Генерация разбиения на слагаемые числа 4

Задача для самостоятельного решения

Как-то раз, придя домой со школы, Света обнаружила записку от мамы, в которой та просила сделать салат. Света знала, что салат — это смесь двух или более ингредиентов, поэтому ей не составило труда выполнить мамину просьбу.

Но Света хочет стать математиком, поэтому для тренировки решила посчитать, сколько различных салатов она сможет сделать из имеющихся продуктов (майонез, огурцы, помидоры). После небольших расчётов она получила ответ: 4.

Зная, что вы любите интересные задачки и хотите стать программистами, Света предлагает вам написать программу, которая определяет количество различных салатов для произвольного числа ингредиентов.

ФИЗИКА

Применение фокусов на занятиях по физике

Л.В. Горбанева,
ст. преподаватель кафедры физики ФГБОУ ВО ПИ ТОГУ
А.М. Голобоков,
студент 5 курса ФЕНМиИТ ФГБОУ ВО ПИ ТОГУ

Наука часто преподаётся в «сухой» и неинтересной форме. Дети учатся механически запоминать информацию, чтобы сдать экзамен, и не видят связи науки с окружающим миром…
Стивен Хокинг,
английский физик-теоретик, космолог, писатель

Введение

Познавательный интерес позиционирует себя как активная познавательная направленность, связанная с положительным эмоциональным отношением к изучению предмета, с радостью познания, с преодолением трудностей, с созданием успеха.

Перед педагогом возникает вопрос: «Как вызвать у ребёнка устойчивый познавательный интерес к предмету?». То есть, необходимо искать путь, с помощью которого можно, добиваясь полноценного усвоения учащимися материала, обеспечивать развитие их познавательной активности, не допуская при этом особой эмоциональной перегрузки.

Одним из путей наиболее эффективного развития познавательного интереса у учащихся является использование физических фокусов во время занятий и во внеурочное время. Прежде чем рассмотреть этот путь, необходимо разобраться, что будем называть фокусом.

Точного определения понятия «фокус» нет. В обычной жизни, не углубляясь в научное объяснение, под фокусом понимается использование конкретных знаний для проведения манипуляций с предметом, для того чтобы обмануть человека и ввести его в заблуждение.

Примерно в IV веке до нашей эры египтяне использовали различные знания наук, чтобы обожествлять фараона перед людьми. Обманным путём жрецы добивались поклонения людей фараону как Богу, используя солнечный диск и падение лучей в нужную точку. Одно из самых ранних свидетельств датируется приблизительно 1700 годом до нашей эры. На древнеегипетском папирусе был изображён некий Деди из Дедснефу, выполняющий фокус перед фараоном.

Древние греки и римляне приходили в восторг от разных трюков, особенно от тех, в которых использовались всевозможные скрытые механизмы. С помощью таких механизмов жрецы совершали настоящие чудеса: массивные двери храмов открывались сами собой, вино лилось из уст и рук мраморных статуй. Фокус под названием «Стаканы и шарики» был описан римлянином Сенекой ещё в I веке нашей эры, но и по сей день его охотно включают в свой репертуар профессиональные иллюзионисты.

В средневековой Европе ремесло фокусника считалось колдовством, а значит, занятием, караемым смертью. Тем не менее, некоторые фокусники умело подчиняли своей власти и влиянию богобоязненных людей. Они не выступали перед большой публикой — им достаточно было на углу улицы собрать немного народа и показать свои «магические трюки», и слух о маге распространялся по всему городу.

Суть всех фокусов была проста — отвлечь внимание людей. Когда человек отвлечён на что-то другое, будь то просто брошенный в воздух платок, фокусник имел возможность беспрепятственно сделать основную часть фокуса. Как говаривал знаменитый сыщик Шерлок Холмс — герой произведений английского писателя Артура Конан Доила: «Мы видим, но не наблюдаем». Заставить зрителя увидеть только то, что происходит у него на глазах, оставляя скрытым тайный механизм трюка, — заветная цель любого фокусника.

Фокусники того времени использовали законы физики, сами того не подозревая (т.к. о них не было ещё известно). Поэтому учёные были заинтересованы фокусами и активно посещали данные мероприятия. Но главная цель была — не разоблачение фокусника, а понять принцип работы фокуса. И раз уж всем известно, что фокусы подчиняются законам физики, их можно смело называть «физическими фокусами».

Использование во время занятий физических фокусов, особенно если они яркие и насыщенные, позволит педагогу вовлечь учащихся в учебный процесс через создание проблемной ситуации, пути решения которой дети и педагог будут искать совместно.

Включать физический фокус в занятие целесообразно на этапе обобщения и закрепления нового материала: педагог демонстрирует фокус, создаёт проблемную ситуацию, учащиеся высказывают своё мнение о произошедшем и, как итог, пробуют связать фокус с пройденным материалом.

На школьных занятиях для демонстрации фокусов педагогу предоставляется немного времени. Поэтому можно вести курс «Физические фокусы» на занятиях в системе дополнительного образования, а также во внеурочной и факультативной деятельности.

В качестве примера в данной статье приводится описание нескольких физических фокусов, которые направлены не на обман и отвлечение внимания учащихся, а на развитие интереса, на возникновение у них вопросов: «Как?» и «Почему?».

Физический фокус, связанный с атмосферным давлением

Учащимся даётся задача: как достать монету из тарелки, не замочив при этом рук? Ответ становится простым, если использовать знание темы «Атмосферное давление».

Нам понадобятся:

– пластилин;

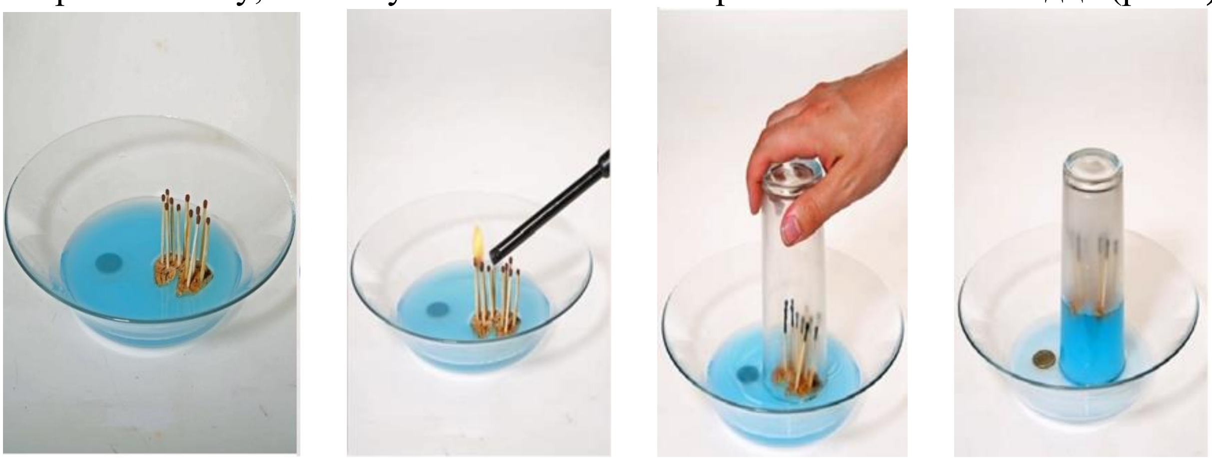
– спички;

– цилиндрический сосуд небольшого объёма.

Последовательность выполнения фокуса

1) В неглубокое блюдце кладём монету и заливаем водой. В пластилин втыкаем спички (5–6 штук). Использовать лучше пластилин, потому что он не плавает на поверхности, и спички легко втыкаются в него.
2) Поджигаем спички, воткнутые в пластилин.
3) Накрываем сверху цилиндрическим сосудом.
4) Ждём, когда вся вода под действием атмосферного давления «вдавится» вовнутрь цилиндрического сосуда.

Забираем монету, оставшуюся лежать на поверхности нашего блюдца (рис.1).



1) 2) 3) 4)

Рисунок 1. Процесс демонстрации фокуса

Объяснение фокуса

Происходит всасывание воздуха, потому что, когда мы зажигаем спички и накрываем цилиндрическим сосудом источник огня, то огонь нагревает воздух внутри. Повышение температуры воздуха влечёт за собой повышение давления и, собственно, объёма воздуха. Но воздух «выгорает», а также уходит из-под стакана. В результате отсутствия кислорода огонь затухает. Внутри стакана температура понижается, следовательно, и понижается давление. Так как внутри сосуда давление мало по сравнению с атмосферным, то в результате атмосферное давление «загоняет» воду под стакан.

Физический фокус «Яйцо в бутылке»

Физический фокус «Яйцо в бутылке» основан на приведённом выше физическом явлении, достаточно прост и почти не требует никаких затрат. Но, несмотря на это, он довольно-таки интересный, и его должен провести каждый начинающий «волшебник».

Для проведения фокуса потребуются:

– сваренное вкрутую и очищенное от скорлупы куриное яйцо среднего размера;

– стеклянная бутылка из-под сока с достаточно широким горлышком;

– полоска бумаги;

– спички или зажигалка;

– растительное масло.

Внимание! Для успешного проведения эксперимента необходимо, чтобы яйцо было не намного больше горлышка бутылки.

Последовательность выполнения фокуса

1) Смажьте горлышко бутылки растительным маслом.

2) Подожгите бумагу и быстро опустите её в бутылку. Будьте осторожны при этом, чтобы не обжечь пальцы!

3) После этого сразу же положите яйцо на горлышко бутылки.

4) Через секунду горящая бумага потухнет, а яйцо невероятным образом окажется в бутылке.

Объяснение фокуса

Горящая бумага нагревает воздух в бутылке, от чего молекулы воздуха приходят в движение, начинают отталкиваться друг от друга. Часть воздуха выходит наружу через щели между яйцом и горлышком бутылки. Когда пламя гаснет, молекулы воздуха охлаждаются и начинают притягиваться друг к другу. Это явление в науке носит название ***парциальный вакуум***. Воздух снаружи бутылки устремляется внутрь неё, однако путь ему преграждает яйцо. Давление молекул воздуха снаружи бутылки настолько велико, что они буквально вталкивают яйцо внутрь сосуда.

Физический фокус «Прокалывание стеклянного пузырька»

Фокус проводится обычно перед объяснением материала. Его показывал известный педагог Анатолий Кузьмич Атаманченко при объяснении темы «Атмосферное давление».

Нам понадобятся:

– небольшой пузырёк с герметичной крышкой;

– надфиль;

– вода;

– тонкая проволока.

Подготовка к фокусу

1) Необходимо надфилем протачить в ребре дна пузырька маленькое отверстие. Это потребует некоторого терпения.

2) Наливаем в пузырёк воду — вода не выливается.

Последовательность выполнения фокуса

1) Пузырёк с водой ставим на демонстрационный стол.

2) Втыкаем в дно тонкую проволоку и показываем детям на просвет, что она внутри пузырька. Потом вытаскиваем её.

Кто-то говорит, что там отверстие. Другие возражают: «Вода бы выливалась». Но после изучения материала все начинают понимать секрет фокуса. Дети с удовольствием повторяют фокус дома.

Физический фокус «Фонтан в пробирке»

Для проведения фокуса потребуются:
– пробирка;
– эпоксидный клей с цементом;
– наконечник от старого стеклянного шприца;
– отрезок трубки;
– отрезок проволоки;
– кювета с водой.

Последовательность выполнения фокуса

1) В пробирку вставляем на эпоксидном клее с цементом наконечник от шприца, на который надет отрезок трубки.

2) Заранее откачиваем воздух и трубку плотно заворачиваем, надевая проволочное кольцо.

3) В нужный момент показываем пробирку, как самую обычную, маскируя завёрнутый конец.

3) Затем незаметно снимаем проволочное кольцо и, опуская конец трубки в кювету с водой, показываем фонтан.

Выглядит это довольно эффектно. Конечно, учащиеся рано или поздно догадываются, что воздух из пробирки был откачан. Но это и замечательно.

Физический фокус «Исчезновение воды в журнале»

Этот фокус очень эффектен и всегда поражает учащихся.

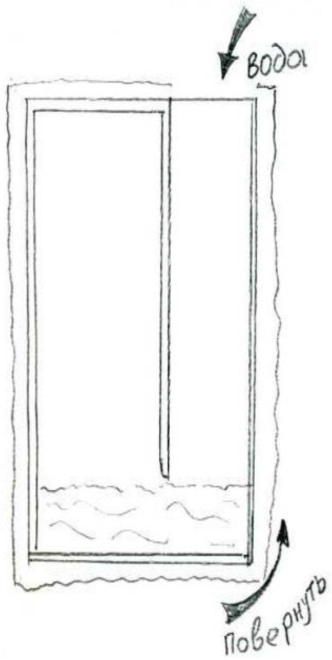
Нам понадобятся:
– журнал;
– вода;
– полиэтилен;
– аппарат для склеивания полиэтилена (утюг, паяльник, выжигатель, аппарат для упаковки).

Подготовка к фокусу

Потребуется время.

1) Сначала берём журнал, в который будем «заливать» воду. Учебник (хоть это и выглядело бы максимально достоверно) лучше не брать — твёрдая обложка мешает нащупывать заливное отверстие.

2) Из упаковочного пакета или лучше из двойного полиэтилена потолще (например, плёнки для теплиц) делаем полиэтиленовый карман размером несколько меньше страницы (чтобы потом его вклеить между страницами). Склеить пленку нужно термическим способом (краем утюга, паяльником, выжигателем, аппаратом для упаковки), соблюдая противопожарные меры. Сначала надо пропаять контур, оставляя небольшое отверстие, затем — перегородку.



Последовательность выполнения фокуса

1) В нужный момент занятия берём журнал из стопки подобных, якобы случайным образом. Показываем его, бегло пролистав.

2) Затем, продолжая говорить, незаметно нащупываем в торце, ближе к переплёту, заливное отверстие. Вставляем туда палец и расширяем его.

3) Заливаем в журнал, якобы между страниц, воду.

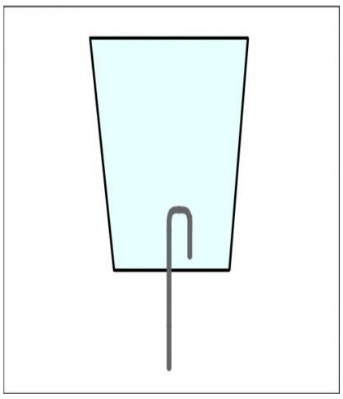
4) Медленно поворачиваем журнал (при этом вода переливается в другое отделение, за перегородку).

Случилось чудо, глаза отказываются верить — вода не выливается.

Несмотря на свою простоту, этот фокус вызывает трудности с объяснением. Учащиеся часто не могут нащупать связь изученной темы и показанного фокуса. Лучше его показать после изучения материала и дать детям подумать над ним дома, так как дома никто не торопит, подключаются папы и мамы, всем интересно.

Физический фокус «Бездонный стакан»

Нам понадобятся:
– непрозрачный стаканчик (вполне подойдёт стаканчик из-под йогурта);
– вода;
– гибкая гофрированная трубочка от сока;
– суперклей или «холодная сварка»;
– сосуд.



Подготовка к фокусу

1) Проделайте в дне стаканчика отверстие.

2) Плотно вставьте в отверстие трубочку, при этом лучше использовать суперклей или «холодную сварку».

3) Верхний конец трубочки согните, как можно ниже ко дну.

Последовательность выполнения фокуса

1) Возьмите стаканчик и налейте в него 10–20 мл воды. Вода при этом выливаться не будет.

2) Повторите процедуру ещё раз и ещё раз. Вода выливаться не будет.

3) Однако после очередного наливания вдруг, неожиданно для зрителей, из стаканчика начнёт выливаться (в заранее приготовленный сосуд) практически вся вода

Объяснение фокуса

Всё дело во внутреннем сифоне. Стаканчик будет «держать» воду до уровня изгиба трубочки, а при его превышении сифон освободит стакан. Поэтому варьируя положение изгиба, можно легко менять «ёмкость» стакана.

Заключение

Применение опытов на занятиях по физике, которые мы преподносим учащимся как фокусы, имеет огромный образовательный и воспитательный потенциал. Во время изучения учащимся предоставляется возможность не только разбираться в показанных педагогом фокусах, но и самим по описанию проводить демонстрацию физических фокусов, а также разрабатывать и демонстрировать их на различных мероприятиях. Трудно переоценить значение такой ситуации, когда учащийся искренне стремится объяснить увиденное на основе знаний по предмету. Это, разумеется, может создавать и проблемную ситуацию, когда знаний недостаточно и нужно искать ответ в различных источниках.

Огромный плюс использования физического фокуса при обучении заключается в том, что он может оставить глубокий эмоциональный след в памяти ребёнка, и, как результат, физика не будет казаться сложным предметом в изучении, а наоборот, очень интересным.

Библиографический список

1. Горев Л.А. Занимательные опыты по физике в 6–7 классах средней школы. – М.: Просвещение, 1985. – 175 с.
2. Ланге В.Н. Экспериментальные физические задачи на смекалку. – М.: Наука, 1985. – 128 с.
3. Тульчинский М.Е Занимательные задачи-парадоксы и софизмы по физике / М.Е. Тульчинский. – М.: Просвещение, 1971. – 160 с.
4. Уокер Дж. Физический фейерверк / пер. с англ. Под ред. И.Ш. Слободецкого. – М.: Мир, 1988. – 298 с.
5. Физические фокусы [Электронный ресурс]. – Режим доступа: http://shkola-fiziki.ru/fizicheskie-fokusy
6. Краткая история фокусов [Электронный ресурс]. – Режим доступа: http://www.micromagic.ru/content/view/24/209/
7. Теорема Менелая [Электронный ресурс]. – Режим доступа: https://foxford.ru/wiki/matematika/teorema-menelaya

Краевое государственное
автономное образовательное учреждение
дополнительного образования
«Центр развития творчества детей (Региональный модельный центр
дополнительного образования детей Хабаровского края»

680000, г. Хабаровск, ул. Комсомольская, 87
тел. / факс: (4212) 30-57-13
Инстаграм: @dop.obrazovanie27
e-mail: yung_khb@mail.ru
http://www.kcdod.khb.ru

Тираж: 25 экз.