Государственное бюджетное общеобразовательное учреждение Московской области

«Сергиево-Посадский физико-математический лицей»

«Внеурочная деятельность – одна из составляющих

профильного изучения информатики

и подготовки к олимпиадам.

Работа с конейнеромmapв С++»

мастер – класс, выступление на семинаре

составитель:

Перлова Наталья Васильевна

учитель информатики

высшей квалификационной категории

**2023 год**

Тема нашего семинара:«Внеурочная деятельность – одна из составляющих профильного изучения информатики и подготовки к олимпиадам. Работа с конейнеромmapв С++».

В лицее ведется обучение в 9-11 классах.Программа базового курса для 9 класса написана на основе «Программа курса «Информатика и ИКТ» для 8 класса и 9 класса. Авторы: Семакин И.Г., Залогова Л.А., Русаков С.В., Шестакова Л.В.

Планирование курса «Информатика и ИКТ» рассчитано на 68 часов (2 часа в неделю в 9 классе).  Информатика изучается в 9 классе.

Для 10-11 классов рабочая программа составлена на основе Программы полного общего образования по предмету «Информатика» (углублённый уровень), разработанной автором учебника, К.Ю. Поляковым, Е.А. Ереминым.

Курс рассчитан на изучение в 10 – 11 классах информационно-технологического профиля обучения общеобразовательной средней школы в течение 34 учебных недель в году общим объемом 136 учебных часов(из расчета 4 часа в неделю).

Внеурочная деятельность ведется во всех параллелях. Мы сегодня познакомимся с материалами курса: «Программирование и математические основы информатики». Данный курс предназначен для учащихся 11 классов и расчитан на 34 часа (один раз в нделю).

Курс ориентирован прежде всего на получение фундаментальных знаний, навыков и умений в области информатики.

***Цели :***

* **освоение и систематизация знаний**, относящихся к математическим объектам информатики;
* **овладение умениями** строить математические объекты информатики, в том числе логические формулы и программы на формальном языке, удовлетворяющие заданному описанию; создавать программы на языке программирования по их описанию; использовать общепользовательские инструменты и настраивать их для нужд пользователя;
* **развитие** алгоритмического мышления, способностей к формализации, элементов системного мышления;
* **воспитание** культуры проектной деятельности, в том числе умения планировать, работать в коллективе; чувства ответственности за результаты своего труда, используемые другими людьми; установки на позитивную социальную деятельность в информационном обществе, недопустимости действий, нарушающих правовые и этические нормы работы с информацией;
* **приобретение опыта** создания, редактирования, оформления, сохранения, передачи информационных объектов различного типа с помощью современных программных средств; построения компьютерных моделей, коллективной реализации информационных проектов, преодоления трудностей в процессе интеллектуального проектирования, информационной деятельности в различных сферах, востребованных на рынке труда.

***Задачи:***

* систематизировать подходы к изучению предмета;
* сформировать у учащихся единую систему понятий, связанных с созданием, получением, обработкой, интерпретацией и хранением информации;
* научить пользоваться распространенными прикладными пакетами;
* показать основные приемы эффективного использования информационных технологий;
* сформировать логические связи с другими предметами, входящими в курс общего образования;
* подготовить учащихся к жизни в информационном обществе.

Основным результатом обучения является достижение базовой информационно-коммуникационной компетентности учащегося.

С поставленными задачами мы успешно справляемся. Результатом является устойчивый интерес учащихся к научно-исследовательской деятельности, вовлечение все большего количества школьников в олимпиадное и исследовательское движение.

Все ученики принимают участие во ВсОШ на школьном уровне.

Результаты ВсОШ 2022- 2023 учебный год:

Муниципальный уровень

«Информационная безопасность» - 10 призеров;

«Информатика» - 2 победителя,12 призеров;

«Робототехника» - 6 победителей.

Региональный уровень

«Информатика» -1 победитель, 1 призер,

6 участников;

«Робототехника» - 3 призера, 1 участник.

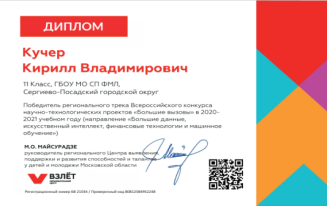
Федеральный уровень

«Информатика» - 1 участник;

«Робототехника» - 2 участника.

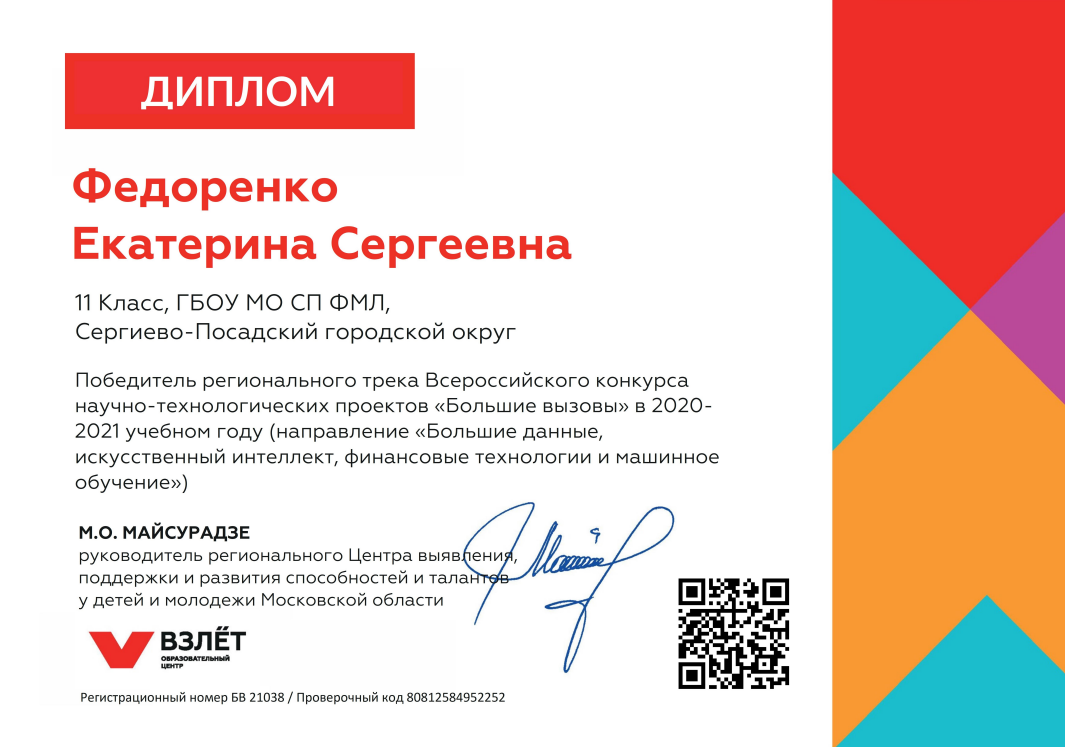
Учащиеся принимают активное участие в различных конкурсах, конференциях и вузовских олимпиадах различного уровня.

Данные 2020-2021 учебного года:

Всероссийский конкурс научно – технологических проектов «Большие вызовы»

Профиль большие данные. «Исследование модели искусственного интеллекта по компьютерному зрению(CycleGAN)на различных данных»

Кучер Кирилл 11 класс - победитель регионального трека

региональный этап, направление: «Большие данные, искусственный интеллект, финансовые технологии и машинное обучение»

Федоренко Екатерина - победитель

XIV Международный конкурс «Математика и проектирование»

«Расчет траектории полета снаряда, от баллисты до БД-13 «Катюши» представляли Молотков Федор, Соловьёв Илья, Ушаков Антон, ученики 9 класса (участники)

Всероссийский конкурс научно – технологических проектов «Большие вызовы»

В направлении «Большие данные, искусственный интеллект, финансовые технологии и машинное обучение»

Щур Виктория -победитель

Итак, вернемся к теме нашего курса: **«Программирование и математические основы информатики».**

В качестве языка программирования используется С++. Почему мы выбрали данный язык? При разработке языка С++ преследовалась цель создания инструментального средства, предназначенного для создания программ разной степени сложности. Язык С++ ориентирован на удовлетворение потребностей программистов. Он предоставляет доступ к аппаратным средствам и позволяет оперировать отдельными битами оперативной памяти. Он включает широкий набор операторов, позволяющих программисту выражать свои идеи в компактном виде. Чаще всего, победители олимпиад по программированию пишут на языках Си и С++ (статистика олимпиад МФТИ).

Вашему вниманию представляется фрагмент занятия «Работа с контейнером «map»». Мне будут помогать ученики.

Рассмотрим задачу ЕГЭ, номер 24:

Текстовый файл содержит только заглавные буквы латинского алфавита (ABC…Z). Определите символ, который чаще всего встречается в файле сразу после буквы A.

Например, в тексте ABCAABADDD после буквы A два раза стоит B, по одному разу — A и D. Для этого текста ответом будет B.

Для решения данной задачи необходимо решить, как мы будем хранить данные о количестве повторов какой либо буквы. Ваши предложения?

(Предполагаемые ответы:

1. Хранить в двух массивах. В одном встречающиеся буквы, в другом, на тех же номерах , сколько раз данная буква встречается.

2. Номер элемента массива – код ASCII буквы, значение сколько раз встречается.)

В С++ есть специальное средство для для программирования задач такого типа- список.

**Список** или контейнер «map»– это упорядоченный набор элементов одного типа, для которого введены операции вставки (включения) и удаления (исключения). **Map** («отображение») – это словарь (ассоциативный массив),  который работает по принципу — [ключ — значение]. Ключ или индекс элемента – любые данные. При добавлении нового элемента контейнер будет отсортирован по возрастанию.

Создание контейнера:

**#include <map>** - подключаем библиотеку;

**...**

**map <char,int> otv; -** объявляем контейнер; в угловых скобках через запятую

задается тип ключа( в нашем случае символ) и тип значения ( число- сколько раз встречается данный символ)

Тогда увеличение счетчика символов - otv[с] ++;

команда добавления элемента в контейнер в общем виде:

**otv.insert ( pair <char,int> (c,1) );**

**алгоритм нашей обработки:**

*вводим первый символ с1*

*пока не конец файла*

*ввести очередной символ с*

*если с== ‘A’ то otv[с] ++*

*с1=с - переход к следующей паре*

*конец пока*

**на языке С++:**

fin>>c1;

while ( !fin.eof())

{ char c;

fin >> c;

if (c1 == 'A') otv[c]++;

c1 = c;

}

Для перебора элементов «map» используется **итератор**. Итератор (или курсор) – специальный объект, который позволяет перебрать все элементы контейнера.

объявление:

**map <char,int>::iterator it;**

С помощью его можно использовать две операции:

* чтобы обратится к **ключу** нужно сделать так: **it->first**
* чтобы обратится к **значению** ячейки нужно сделать так: **it->second.**

Вывод на экран элементов нашего контейнера будет выглядеть следующим образом:

**for (it=otv.begin();it<otv.end();it++)**

**{cout << it->first << " " << it->second << endl;}**

Второй способ:

**for (pair<char, int> p : otv)**

**{cout << p.first << " " << p.second << endl;}**

Таким образом решение нашей задачи:

**#include <iostream>**

**#include <fstream>**

**#include <map>**

**using namespace std;**

**int main()**

**{ map<char, int> otv;**

**fstream fin;**

**fin.open("gg.txt");**

**char c1;**

**fin >> c1;**

**while (!fin.eof())**

**{ char c;**

**fin >> c;**

**if (c1 == 'A') otv[c]++;**

**c1 = c;**

**}**

**for (pair<char, int>p : otv)**

**cout << p.first << " " << p.second << endl;**

**fin.close();**

**system("pause"); }**

Для вас подобраны олимпиадные задачи, которые решаются с использованием контейнера «map».

Раздаточный материал:

Система регистрации

ограничение по времени на тест

5 seconds

ограничение по памяти на тест

64 megabytes

ввод

стандартный ввод

вывод

стандартный вывод

В скором времени в Берляндии откроется новая почтовая служба "Берляндеск". Администрация сайта хочет запустить свой проект как можно быстрее, поэтому они попросили Вас о помощи. Вам предлагается реализовать прототип системы регистрации сайта.

Система должна работать по следующему принципу. Каждый раз, когда новый пользователь хочет зарегистрироваться, он посылает системе запрос name со своим именем. Если данное имя не содержится в базе данных системы, то оно заносится туда и пользователю возвращается ответ OK, подтверждающий успешную регистрацию. Если же на сайте уже присутствует пользователь с именем name, то система формирует новое имя и выдает его пользователю в качестве подсказки, *при этом подсказка также добавляется в базу данных*. Новое имя формируется по следующему правилу. К name последовательно приписываются числа, начиная с единицы (name1, name2, ...), и среди них находят такое наименьшее *i*, что name*i* не содержится в базе данных сайта.

**Входные данные**

В первой строке входных данных задано число *n* (1 ≤ *n* ≤ 105). Следующие *n* строк содержат запросы к системе. Каждый запрос представляет собой непустую строку длиной не более 32 символов, состоящую только из строчных букв латинского алфавита.

**Выходные данные**

В выходных данных должно содержаться *n* строк — ответы системы на запросы: ОК в случае успешной регистрации, или подсказку с новым именем, если запрашиваемое уже занято.

Напрмер:

|  |  |
| --- | --- |
| входные данные | выходные данные |
| 4  abacaba  acaba  abacaba  acab | OK OK abacaba1 OK |
| 6 first first second second third third | OK first1 OK second1 OK third1 |

Футбол

ограничение по времени на тест

2 seconds

ограничение по памяти на тест

256 megabytes

ввод

стандартный ввод

вывод

стандартный вывод

Однажды Вася решил посмотреть результаты финала чемпионата Берляндии по футболу 1910 года. К сожалению, он не нашел итоговый счет матча, зато нашел подробное текстовое описание хода всей игры. Всего в этом описании оказалось *n* строк, каждая из которых описывала один гол. Каждый гол обозначался названием забившей его команды. Помогите Васе — узнайте название команды, победившей в финале. Гарантируется, что матч не закончился вничью.

**Входные данные**

В первой строке записано целое число *n* (1 ≤ *n* ≤ 100) — количество строк в описании. Далее следует *n* строк — для каждого мяча записано название забившей его команды. Названия — непустые строки из больших латинских букв, длины которых не превосходят 10 символов. Гарантируется, что матч не закончился в ничью, и в описании встречается не более двух различных команд.

**Выходные данные**

Выведите название победившей команды. Напоминаем, что в футболе победившей считается команда, забившая больше мячей.

Напрмер:

|  |  |
| --- | --- |
| входные данные | выходные данные |
| 1 ABC | ABC |
| 5 A ABA ABA A A | A |

Миша и смена хэндлов

ограничение по времени на тест

1 секунда

ограничение по памяти на тест

256 мегабайт

ввод

стандартный ввод

вывод

стандартный вывод

Взломав сайт Codeforces, Миша решил дать возможность всем пользователям менять их хэндлы. Пользователь теперь может сменить свой хэндл сколько угодно раз. Но при этом каждый новый хэндл не должен совпадать ни с каким из уже занятых или занятых в прошлом хэндлов.

У Миши есть список запросов пользователей на смену хэндлов. После их выполнения он хочет понять соответствие между исходными хэндлами пользователей и новыми. Помогите ему в этом.

**Входные данные**

В первой строке находится целое число *q* (1 ≤ *q* ≤ 1000), количество запросов на смену хэндла.

В последующих *q* строках находится описание запросов, по одному в строке.

Каждый запрос состоит из двух непустых строк *old* и *new*, разделенных пробелом. Строки состоят из заглавных и прописных символов латинского алфавита и цифр. Строки *old* и *new* различны. Длины строк не превосходят 20.

Запросы даны в хронологическом порядке. Иными словами, к моменту очередного запроса, существует единственный человек с хэндлом *old*, а хэндл *new* никем не используется и не был никем использован до этого.

**Выходные данные**

В первой строке выведите целое число *n* — количество пользователей, воспользовавшихся возможностью сменить хэндл.

В последующих *n* строках выведите соответствие между старыми хэндлами пользователей и новыми. Каждая из них должна содержать по две строки *old* и *new*, разделенные пробелом, что означает следующее: до взлома сайта пользователь имел хэндл *old*, а после выполнения всех запросов получил хэндл *new*. Строки разрешается выводить в любом порядке.

Каждый пользователь, менявший хэндл, должен встретиться в этом описании ровно один раз.

Напрмер:

|  |  |
| --- | --- |
| входные данные | выходные данные |
| 5 Misha ILoveCodeforces Vasya Petrov Petrov VasyaPetrov123 ILoveCodeforces MikeMirzayanov Petya Ivanov | 3 Petya Ivanov Misha MikeMirzayanov Vasya VasyaPetrov123 |

Репосты

ограничение по времени на тест

1 секунда

ограничение по памяти на тест

256 мегабайт

ввод

стандартный ввод

вывод

стандартный вывод

Однажды Поликарп опубликовал в социальной сети смешную картинку с опросом про цвет своего хэндла. Многие из его друзей стали репостить шутку Поликарпа себе в ленту. Некоторые из них репостили репосты и так далее.

Эти события заданы в виде последовательности строк «*name1* reposted *name2*» где *name1* — это имя того, кто репостнул, а *name2* — имя того, с чей ленты репостнули шутку. Гарантируется, что для каждой строки «*name1* reposted *name2*» пользователь «*name1*» еще не имел эту шутку в свой ленте, а «*name2*» уже имел ее в своей ленте к моменту репоста. Поликарп зарегистрирован под именем «Polycarp», и изначально шутка есть только в его ленте.

Поликарп измеряет успешность шутки как длину наибольшей цепочки репостов. Выведите успешность шутки Поликарпа.

**Входные данные**

В первой строке входных данных записано целое число *n* (1 ≤ *n* ≤ 200) — количество репостов. Далее записаны сами репосты в порядке их совершения. Каждый из них записан в отдельной строке и имеет вид «*name1* reposted *name2*». Все имена во входных данных состоят из прописных или строчных латинских букв и/или цифр и имеют длины от 2 до 24 символов включительно.

Известно, что имена пользователей регистронезависимы, то есть два имени, отличающиеся исключительно регистром букв, соответствуют одному и тому же пользователю соцсети.

**Выходные данные**

Выведите единственное целое число — максимальную длину цепочки репостов.

Напрмер:

|  |  |
| --- | --- |
| входные данные | выходные данные |
| 5 tourist reposted Polycarp Petr reposted Tourist WJMZBMR reposted Petr sdya reposted wjmzbmr vepifanov reposted sdya | 6 |
| 6 Mike reposted Polycarp Max reposted Polycarp EveryOne reposted Polycarp 111 reposted Polycarp VkCup reposted Polycarp Codeforces reposted Polycarp | 2 |
| 1 SoMeStRaNgEgUe reposted PoLyCaRp | 2 |