МОЙ ПУТЬ В С++

12.02 Мой путь в С++ начался с изучения книги, однако теория в программировании - ничто, по сравнению с практикой. Начав решать задачи на платформе, я поняла принципы и основы. На решение самых элементарных задач у меня уходит много времени, так как мне нужно много раз внимательно перечитать условие, чтобы осознать, чего от меня хотят. Буду продолжать решать задачи в Stepik!!

13.02 – 18.02 Задачи с числами довольно легкие, решаю по 1-2 задачи каждый день. Начала разбирать задачи по теме «Условный оператор», все идет довольно хорошо, но нужно определенное количество времени, чтобы понять условие.

19.02 Сегодня я наконец разобрала задачи на шахматы, над которыми думала все выходные.

20.02 – 25.02 Пытаюсь понять как решать задачи на шахматы с использованием функций, очень тяжело понять принцип. Параллельно решаю по 1 задаче в день по предыдущим темам.

26.02 Снова вернулись к шахматам и решали их другим способом. Написали код, который позволяет решить все 5 типов задач.

26.02 Сбила свой рекорд в 21 день….

27.02 Решила все задачи с использованием условного оператора, повторила принципы их решения

28.02 Взялась основательно за цикл while, посмотрела видеоурок, чтобы точно вспомнить

29.02 Решаю задачи на цикл while, на решение 3 задач у меня ушел целый час((

01.03 Сегодня задачи на while посложнее(( без подсказок в комментариях не справляюсь, на решение 2-х задач у меня ушло около 40 минут…

02.03 Сегодня изучаю действительные числа, решаю задачи в степике на эту тему. Задачи решаются довольно легко. На решение одной задачи уходит 15-20 минут.

03.03 Пробую решать задачи на цикл for и массивы, практикую использование вектора в задачах. Сначала было тяжело понять, но после - задачи стали решаться относительно несложно. Поняв, что процесс считывания массива из задачи в задачу не меняется, мне стало проще, так как оставалось менять лишь малую часть код под условие задачи. За 1 час я решила 4 (!!) задачи. Буду продолжать.

04.03 – 06.03 Учусь использовать функции и начну решать задачи по теме. Решаю по одной задаче в день.

07.03 Решила 2 задачи на использование функций.. Решение до меня очень долго доходит((

04.04 После долгого перерыва в обучении я буду снова возвращаться в прежний ритм, чтобы добить задачи! Сегодня я вспоминала, как работать с циклами for и массивами. Мне потребовалось пересмореть видеоурок, перечитать лекции, чтобы выполнить задания по теме. На решение 4 задач у меня ушло почти 3 часа (с перерывами). Вспомнила функции b.push\_back(),b.size(), их назначение.

05.04 Сегодня окончательно разделаюсь с массивами, чтобы наконец закрепить данную тему. На решение 5 задач у меня ушло 3,5 часа. Некоторые задачи были схожи по условию.

06.04 Сегодня решаю блок функция и рекурсия вспомнила уравнение ромба |x| + |y| = r, для того чтобы решить задачу. Функция pow() нужна для возведения в степень.

Power() – рекурсивная фукнция (вызывает сама себя)

На решение 5 задач у меня ушло около 2 часов, однако без подсказок я бы не справилась.

07.04 Fact() – тож рекурсивная.

Пересмотрела урок по функциям и рекурсии, продолжаю решать задачи на данную тему.

Решила две задачи по теме, дальше как-то не получается… На решение двух задач без учета видеоурока у меня ушло 30 минут (ошибки были по причине того, что тип данных я указывала int, а не double, где это было нужно)

Setprecision() – функция, которая определяет точность вывода нецелого числа после запятой

Вернулась к теме действительные числа, решила 2 задачи на квадратное уравнения. Они были почти одинаковыми. (Решила за 40 минут)

09.04 Строки и символы.

Getline( , ) – зачитывает всю строку до перевода на следующую

Метод find() находит подстроку в строке. Если не находит, возвращает -1.

На решение 6 задач у меня ушло 2 часа вместе с просмотром видеоурока. После предыдущих задач эти кажутся супер-простыми!!!!

10.04 Сегодня решаю задачи на тему множества (пересекающиеся кружочки) #include <set> и словари.

Словарь в С++ - библиотека map #include <map> ( s[ ] = “”) похоже на вектор

Lower\_bound возвращает указатель на первый элемент, значение которого больше либо равно переданному параметру.

Upper\_bound — на первый элемент, который строго больше.

Set – подсчитывает число различных элементов в последовательности.

Удаление элемента – s.erase()

Структура multiset, в отличие от set, может хранить одинаковые элементы.

15.04 Yandex Handbook.

Аргументы функций: **аргументы сложных типов (векторы, строки, любые контейнеры, большие структуры) всегда лучше передавать в функцию по константной ссылке, если функция использует их только для чтения.**

**16.04 -18.04 Чтобы все вспомнить, смотрю курс видеоуроков.**

**Изображение выглядит как текст, программное обеспечение, Веб-сайт, веб-страница

Автоматически созданное описание**

20.04 – 23.04 Продолжаю смотреть.

24.04 Изучаю ассоциативные контейнеры и как их использовать во втором проекте по программированию.

Работа с диманическим массивом. (Арендуем память)Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, Графическое программное обеспечение

Автоматически созданное описание

Map – хранит только уникальные ключ, а Multimap – может хранить и повторяющиеся ключи тоже.

27-28.04 Пробую решать задачи в хендбуке на последовательные контейнеры.

**range-based for:**

for (const auto& elem : data) {

// ...

}

28.04 Yandex Handbook.

Аргументы функций: **аргументы сложных типов (векторы, строки, любые контейнеры, большие структуры) всегда лучше передавать в функцию по константной ссылке, если функция использует их только для чтения.**

**Пробую решать задачи в хендбуке на последовательные контейнеры.**

**range-based for:**

**for (const auto& elem : data) {**

**// ...**

**}**

17.05 Так как мы активно работаем над вторым проектом, я немного забросила практику. Буду восстанавливаться!

Сегодня решила одну задачу в Яндекс Хендбуке на последовательные контейнеры.

Вторая задача была довольно легкой, так как там требовалось лишь знание о том, как помещать значения в начало / конец контейнера.

Для этого используем push\_front / push\_back.

18.05 Итак, я решила задачу в хендбуке на последовательные контейнеры. Задача была про вагоны поезда.

Суть задачи в том, что:  
Во-первых, нужно понять какой контейнер здесь лучше использовать. Здесь был использован контейнер deque, так как *его итераторы являются итераторами произвольного доступа, а значит, можно будет быстро найти место расцепки.*

Во-вторых, использование pop\_back() и pop\_front() для того, чтобы удалять необходимые нам элементы (в нашем случае, вагоны) сразу столько, сколько нам нужно!!

В-третьих, я открыла для себя использование .empty(), которое позволяет вычислить есть ли что-то в контейнере или он пуст.

Мои ошибки при решении были по невнимательности..

А также проблема была в том, что после того, как я прописала все условия, забыла дописать break, что привело к тому, что функция работала неверно

Изображение выглядит как снимок экрана, текст, программное обеспечение, веб-страница

Автоматически созданное описание

Боюсь, что следующие задачи будут сложными для меня, но буду пробовать!

23.05 Для решения четвертой задачи из яндекс хендбука я использовала контейнер std::list, так как для решения этой задачи нам не нужен произвольный доступ к элементам. (Нужен последовательный)

При написании кода, нужно понять, какие условия стоит учесть для правильного функционирования. Например, при использовании down нужно учитывать, не находится ли курсор на последней строке и тд.

Для учета всех условий я использовала условный оператор if, else if.

Открыла для себя новую функцию std::move, которая не используется без std::

“Обратите внимание, что при вырезании строки в буфер мы используем функцию std::move. Она позволяет забрать владение строкой, которая вот-вот будет удалена из списка, и сэкономить на её копировании.”

24.05 Сегодня я решила одну из задач на ассоциативные контейнеры. Она была довольно простой, так как здесь требовалось лишь знание теории.

Для решения я использовала контейнер set.

Чтобы узнать содержится ли тот или иной элемент во множестве, мы используем .contains()

Для того чтобы вставить элемент в последовательность, мы используем .insert()

Итого я решила 5/5 задач в Хендбуке. По возможности буду и дальше практиковаться для лучшей подготовки к предстоящему экзамену.

Двумерные массивы в С++, Stepik

Прежде чем приступить к ршению задач на степике, я изучила теорию по данной теме. На степике ее оказалось недостаточно и я пошла искать объяснения на просторах интернета. Наткнувшись на первое попавшееся видео, разобрала основы !!

Далее приступила к решеню самих задач на степике.

Первые две задачи на двумерные массивы были довольно легкие!!

Изображение выглядит как текст, программное обеспечение, компьютер, Значок на компьютере

Автоматически созданное описание

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, Значок на компьютере

Автоматически созданное описание

Тут я заметила, что сделала небольшую ошибку в том, что при выводе массива я ошиблась числом. Нужно было либо добавить новый столбец в начальный массив, либо изменить при выводе на “j<4”. Именно поэтому последний столбец был продублирован.

2 задача: Для решения второй задачи мне понадобилась данная таблица.   
До меня почему-то долго доходило, откуда взялась формула побочной диагонали и поэтому возникли некоторые трудности.

Изображение выглядит как текст, линия, Шрифт, диаграмма

Автоматически созданное описание

3 задача: Эта задача лёгкая, но достаточно интересная по своей логике. Она не требует никаких условий, если заметить такую закономерность: Изображение выглядит как снимок экрана, текст, Шрифт, типография

Автоматически созданное описание

**Модуль вычитания номера строки из номера столбца будет равен необходимому нам числу ! ( abs( i – j ) )**

4,5 задачи: Следующие 2 задачи были несложные, на решение двух задач я потратила около 50 минут. (Без подсказок в комментяриях, как сделать, чтобы код выглядел «солиднее» не обошлось)

6. Эта задача отсылает нас к 3 задаче. Изменилась лишь формулировка и вывод. Единственное отличие состоит в том, что нам **не нужно использовать модуль**, тк k может быть отрицательным числом тоже.

Изображение выглядит как снимок экрана, прямоугольный

Автоматически созданное описание

(Пример: 1-0=1 (диагональ лежит ниже), а 0-1=-1 (диагональ-выше)).

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

7 задача: Здесь было довольно легко, так как схема трансплантации матрицы понятна.

Изображение выглядит как текст, Шрифт, линия, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст, программное обеспечение, снимок экрана, Мультимедийное программное обеспечение

Автоматически созданное описание

8 задача: Без подсказок в комментариях я бы не справилась, но в целом задача несложная:

Перед прохождением каждого ряда устанавливаем счетчик равным нулю.

Если встречается 0 => += 1;

Если 1 - счетчик обнуляем;

Затем, если счетчик == k, выводим номер ряда (i +1;), т.к. рядов с номером 0 не бывает, и завершаем программу.

Если ни в одном ряду запрошенного количества мест, находящихся рядом нет, -> выводим 0;

Когда ряд кончается надо проверить не было ли кол-во нулей больше максимума.

Тяжело было именно учесть все условия, чтобы не провалиться на других тестах. На решение данной задачи я потратила час.

9 задача Решение похоже на 7 задачу, но с единственным изменением: выводить столбцы с конца

Данный комментарий с картинкой помог быстрее понять принцип движения массива

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Тут самый подходящий способ (возможно единственный) это такой, что нужно на каждом шаге проверять является ли следующая ячейка не нулём. Если там не ноль, то нужно изменить направление. Перед заполнением двумерного массива, нужно заполнить края единицами, а основное поле нулями. То есть сделать массив на [n + 2][m + 2] ячеек.

В итоге, я сдалась и решила посмотреть в решение последних задач, так как они оказались для меня слишком трудными.

Выводы по изучению курса языка C++:

С++ это мой первый язык программирования, с которого я начала изучение.

Все, что я делала на протяжение эти 4-5 месяцев было для меня в новинку.

Несмотря на то что программирование вызывало много стресса, так как оно мне давалось легко в начале и очень трудно в конце, я рада, что начала изучение с него. Как язык программирования – он очень простой и понятный.

Если говорить о stepik, то сама платформа очень удобная для изучения языка, однако очень не хватает теории и призодится искать ответы на вопросы либо в комментариях к задачам, либо на просторах интернета.

Выделить самую сложную из всех тем не могу, так как первые задачки везде были довольно легкими. Однако последние задачи на «Двумерные массивы» оказались, к сожалению, мне не под силу..

Это был интересный опыт! Теперь я знаю, что летом я серьезно займусь этим языком и буду углублять свои знания!)

Мои заметки:

- Структуры: ! не забывать ставить ; после фигурной скобки в конце

- ++b and b++

- часы на часах n / 3600%24

- если 4-х значное число симметрично, произведение сумм двух первых и двух последних равно нулю