**Основная часть**

**Описание алгоритмов**

**Чтение данных**

Чтение происходит в три этапа. Сначала считываются точки, потом схемы, затем потоки. Каждая читается с помощью регулярных выражений в соответствии с форматом.

Основная идея одинакова для всех трёх файлов. Процедура, соответствующая файлу, получает на вход путь к нему и массив структур, который требуется заполнить. Далее объявляется счётчик, который отвечает за движение по массиву. Затем делается попытка открыть файл. В случае успеха выводится сообщение об открытии файла, иначе выводится сообщение об ошибке, и программа останавливает свою работу. Если файл был открыт, то первым делом происходит считывание длины массива и проверка на не отрицательность этого числа. В случае, если длина меньше либо равна нуля, то происходит остановка программы. После чего длина массива становится равной полученному числу. Далее объявляется регулярное выражение для работы с данными из файла. И начинается основной цикл, в котором происходит заполнение массива. Файл читается построчно и построчно обрабатывается регулярным выражением до тех пор, пока не закончится файл.

Первым делом делается проверка на успешность сопоставления регулярного выражения и строки. В случае неуспеха выводится сообщение об ошибке. Также проверяется уникальность имён входных данных. Далее делается проверка на возможность доступа к текущей координате массива. В случае невозможности осуществить это программа заканчивает работу и предупреждает о нехватке выделенного места под элементы массива. Затем идёт заполнение, подробности которого можно найти ниже. И, наконец, после успешного наполнения массива, делается проверка и правка, чтобы действительная длина массива совпадала с декларируемой, после чего файл закрывается.

**Чтение точек**

Напомним содержимое структуры *checkPoint*. Она состоит из семи полей:

1. Имя точки
2. Координата x
3. Координата y
4. Координата z
5. Минимальная скорость Vmin
6. Максимальная скорость Vmax
7. Флаг посадочной полосы

Согласно формату, в таком же порядке данные находятся в файле. Теперь рассмотрим регулярное выражение:

([a-z,A-Z]\w\*)\s+([-+]?[0-9]\*\.?[0-9]+)\s+([-+]?[0-9]\*\.?[0-9]+)\s+([-+]?[0-9]\*\.?[0-9]+)\s+([0-9]\*\.?[0-9]+)\s+([0-9]\*\.?[0-9]+)\s\*(LAND|0)?\s\*

Первая группа захвата отвечает за имя точки. Согласно формату, имя начинается с буквы. Далее может быть любые комбинации букв и цифр.

Вторая, третья и четвёртая группы захвата одинаковые и предназначены для распознания действительного числа.

Пятая и шестая почти аналогичны предыдущим трём, но они захватывают не все действительные числа, а только неотрицательные.

И, наконец, седьмая ищет совпадение с флагом посадки LAND, нулём либо же отсутствует вовсе. Однако, одна и только одна точка с таким флагом должна быть. За корректность этого отвечает счётчик «k».

Далее происходит заполнение структуры *checkPoints* поэлементно, в зависимости от исходных данных. Поддерживается ввод в следующих единицах измерения:

* Координата в метрах, в километрах и в морских милях
* Скорость в метрах в секунду, километрах в час и морских милях в час

Флаг посадки по умолчанию *false*. *True* присваивается, если седьмая группа захвата содержит *LAND*. И увеличивается счётчик «k». Если он превосходит единицу – программа останавливается и предупреждает об ошибке, что посадочных точек больше одной.

Потом происходит заполнение отображения Имя точки  ID точки.

Затем происходит проверка корректности введённых скоростей, чтобы максимальная скорость не была меньше минимальной. И увеличивается счётчик «i».

Наконец, делается финальная проверка, что если k остался равен нулю, что означает отсутствие точки посадки, тогда программа завершает работу и предупреждает об ошибке.

**Чтение схем**

В файле со схемами лежит информация об обычных схемах и стандартных схемах. Начнём описание с обычных.

Напомним поля структуры *scheme*:

1. Имя схемы
2. ID точки начала схемы
3. Массив точек, которыми схема оканчивается
4. Массив точек, составляющие основной путь схемы
5. Массив точек, откуда можно спрямляться
6. Массив точек, куда можно спрямляться

Рассмотрим регулярное выражение и те части, которые относятся к обычной схеме:

(\w+)\s\*(?:\((\w+)\))?\s\*\(([\w\s]\*|\d+)\)\s\*:(?:\s\*([\w\s]\*)\s+(?:Str\(([\w\s]+)\)\s\*([\w\s]+)\/Str)?\s\*([\w\s]\*)?)?

Первая группа захвата — это имя схемы. Представляет из себя любой набор букв и цифр.

Вторая группа – это имя точки, с которой начинается схема.

Третья – это имена точек, которой оканчивается схема.

Далее после двоеточия следует основное тело схемы. Четвертая группа — это точки схемы, предшествующие спрямлению. Если оно отсутствует, то эта группа представляет из себя путь. Однако, в связи с тем, что после этой группы обязательно должен быть хотя бы один пробел, и если после последней точки схемы пробела не будет, то эта точка не попадёт в эту группу захвата. Нужно искать в седьмой.

Пятая группа – это точки, на которые можно спрямляться с точек из шестой группы. Если спрямление в схеме отсутствует, то эти группы пустые.

Седьмая группа – это точки, идущие после спрямления. Или, если нет спрямления в схеме и нет пробела после последней точки, тогда тут находится та самая последняя точка.

Также для удобства записи схем состоящих из двух точек, было введено упрощение записи. Можно вводить только имя схемы, точку начала, точку конца и двоеточие. Что регулируется большой не захватывающей скобкой после двоеточия с вопросительным знаком на окончании.

Если спрямление присутствует, то путь состоит из четвёртой, шестой и седьмой групп захвата.

Заполнение. В общем случае первая группа сразу же присваивается в первое поле структуры. Второе поле – это результат отображения Имя точки --> ID точки, где в качестве имени выступает вторая группа захвата.

Далее необходимо заполнить массив конечных точек. Для этого используется вспомогательная функция, которая получает строку на вход и массив, который требуется заполнить. Строка читается поэлементно, и с помощью отображения Имя точки --> ID точки заполняет данный массив. В случае, если точки не оказалось среди *checkPoints*, тогда выдаётся ошибка и программа заканчивает свою работу.

Затем абсолютно аналогично заполняются поля пять и шесть. Теперь необходимо собрать путь схемы. Для этого используя ту же вспомогательную функцию с четвёртой группой и полем путь, мы собираем точки до спрямления. Затем записываем в путь все элементы из поля пять, и, наконец, используя вспомогательную функцию с седьмой группой и полем путь, мы собираем точки после спрямления.

Если четвёртая и пятая группы пустые, то это означает, что мы имеем дело со схемой, состоящей из двух точек. В этом случае, имя и точка начала схемы заполняются как в общем случае, но также нам известно, что массив конечных точек состоит из одного элемента, поэтому мы можем его сразу же записывать в массив конечных точек. И тогда путь, также легко определяется, как точка начала схемы плюс первый, он же и единственный, элемент массива конечных точек.

После чего увеличивается счётчик «i».

Теперь опишем работу со стандартными схемами.

Напомним, что стандартной схемой называется схема типа «тромбон». Она определяется тремя точками и количеством повторений, что и отражено в полях структуры *StandardScheme*:

1. Имя схемы
2. ID точки начала и конца стандартной схемы
3. ID точки начала обратного плеча стандартной схемы
4. ID точки конца обратного плеча стандартной схемы
5. Количество повторений стандартной схемы
6. Минимальное время, за которое можно пройти схему
7. Максимальное время, за которое можно пройти схему

Поля шесть, семь вычисляются в дальнейшем.

Рассмотрим регулярное выражение и те части, которые относятся к стандартной схеме:

(\w+)\s\*(?:\((\w+)\))?\s\*\(([\w\s]\*|\d+)\)\s\*:(?:\s\*([\w\s]\*)\s+(?:Str\(([\w\s]+)\)\s\*([\w\s]+)\/Str)?\s\*([\w\s]\*)?)?

Первая группа захвата – это имя стандартной схемы.

Так как начало и конец совпадают, то вторая группа захвата это второе поле структуры.

Третья группа – это количество повторений схемы.

В стандартной схеме нет спрямлений, поэтому пятая и шестая группы всегда пустые. Четвёртая и седьмая вместе дают все три точки. Поэтому с помощью вспомогательной функции из обычных схем, заполняем временный массив *path*, аналогичный одноименному полю из обычных схем. И, затем, заполняем второе, третье и четвёртые поля структуры стандартной схемы как – первый, второй и третий элемент массива *path*. Если же длина этого массива получилась больше трёх, то программа завершает работу и выдаёт ошибку.

Далее заполняется отображение ID точки начала стандартной схемы--> ID стандартной схемы.

После чего увеличивается счётчик «k».

**Чтение потоков**

В отличии от двух предыдущих структур, тут мы не будем рассматривать все поля, так как большая их часть будет заполнена дальше. На данном этапе вспомним, что есть два поля: имя потока и имя первой точки потока.

Рассмотрим регулярное выражение:

(\w+)\s+(\w+)\s\*

Первая группа отвечает за имя потока – любой набор букв и цифр. Вторая за имя первой точки потока.

Первому полю присваивается значение первой группы. Второму – значение отображения Имя точки --> ID точки от второй группы.

После чего увеличивается счётчик «i».