***1.Постановка задачи.***

Основные задачи работы:

* реализация рекурсивного алгоритма переноса колец с выводом номера переносимого кольца и номера хода;
* (псевдо)графическое отображение состояния модели башен (возможно – с анимацией переноса колец);
* анализ изменения времени выполнения алгоритма при изменении числа колец и определение временной сложности алгоритма переноса.

***2.Теория***

Ханойская башня является одной из популярных [головоломок](http://ru.wikipedia.org/wiki/%D0%93%D0%BE%D0%BB%D0%BE%D0%B2%D0%BE%D0%BB%D0%BE%D0%BC%D0%BA%D0%B0) [XIX века](http://ru.wikipedia.org/wiki/XIX_%D0%B2%D0%B5%D0%BA). Даны три стержня, на один из которых нанизаны восемь колец, причем кольца отличаются размером и лежат меньшее на большем. Задача состоит в том, чтобы перенести пирамиду из восьми колец за наименьшее число ходов на другой стержень. За один раз разрешается переносить только одно кольцо, причём нельзя класть большее кольцо на меньшее.

Эту игру придумал французский математик [Эдуард Люка](http://ru.wikipedia.org/wiki/%D0%AD%D0%B4%D1%83%D0%B0%D1%80%D0%B4_%D0%9B%D1%8E%D0%BA%D0%B0), в [1883 году](http://ru.wikipedia.org/wiki/1883_%D0%B3%D0%BE%D0%B4), её продавали как забавную игрушку. Первоначально она называлась «Профессор Клаус (Claus) из Колледжа Ли-Су-Стьян (Li-Sou-Stian)», но вскоре обнаружилось, что таинственный профессор из несуществующего колледжа — не более чем анаграмма фамилии изобретателя игры, профессора Люка (Lucas) из колледжа Сен-Луи (Saint Louis).

Для того чтобы перенести самый большой диск, нужно сначала перенести все диски, кроме последнего, на второй стержень, потом перенести самый большой на третий, после чего останется перенести все остальные диски со второго на третий. Задачу о переносе N-1 диска мы решаем аналогично, только поменяем стержни местами (при первом переносе конечным стержнем будем считать второй, а не третий, при втором переносе начальным вместо первого будет второй). Как видите, задачка разрешима. И правда, ведь задачу о N-1 дисков мы сведем к задаче о N-2 дисков, ту в свою очередь к N-3 дискам, и так вплоть до 1 диска. Также ясно, что это единственно правильный подход, так что жрецы, скорее всего, действуют так же. Этот метод легко программируется с помощью *рекурсии.*

**Рекурсия** — вызов функции из неё же самой (обычно с другими значениями входных параметров), непосредственно или через другие функции (например, функция А вызывает функцию B, а функция B — функцию A). Количество вложенных вызовов функции называется глубиной рекурсии. Мощь рекурсивного определения объекта в том, что такое конечное определение способно описывать бесконечно большое число объектов. С помощью рекурсивной программы возможно описать бесконечное вычисление, причём без явных повторений частей программы.

***3.Руководство программиста***

Это программа разработана на языке С++ и содержит следующие функции.

***void DrawHanoi(int n)*  -** *функция, осуществляющая вывод на экран дисков и стержней. Параметры: количество дисков.*

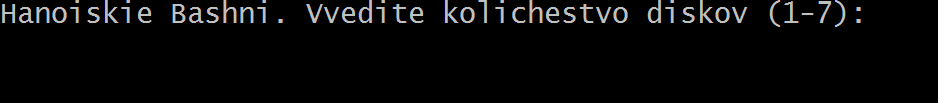
***void MoveDisk(int Disk,int from, int to) -*** *функция, осуществляет анимацию перемещение дисков. Параметры: количество дисков, номер стержня с которого перемещаем, номер стержня на который перемещаем.*

**void Towers(int n,int from,int to,int aux) -**  *рекурсивная функция осуществляет перенос дисков. Праметры: количество дисков, номер стержня с которого перемещаем, номер стержня на который перемещаем, номер промежуточного стержня через которые будем перекладывать диски.*

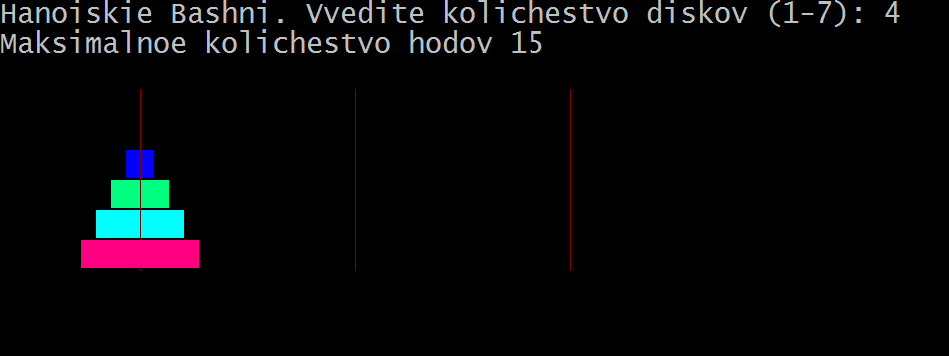
***intmain()* –** *основная функция программы. Инициирует графический режим, запрашивает у пользователя количество дисков, и после выполнения задачи, выводит на экран выполнения функции* ***Towers***

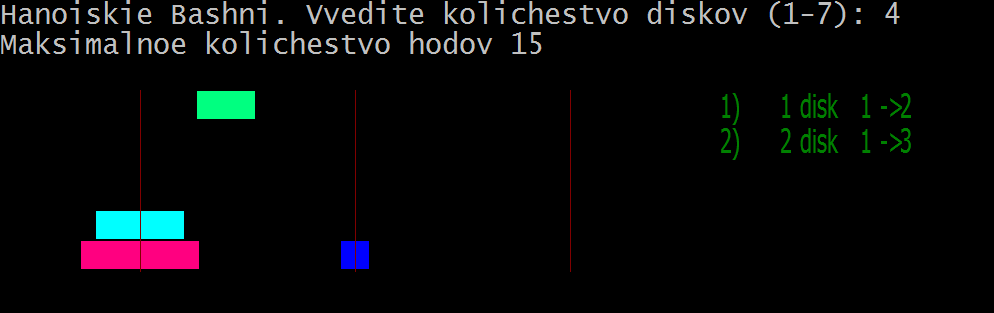
***4.Тестирование***

При запуске программы пользователю предлагается выбрать количество дисков.

******

При нажатии на любую клавишу клавиатуры осуществляется перемещение дисков. В процессе перемещения дисков, выдается программой с какого на какой стержень и какой диск перемещается, а также суммарное количество перемещений.





***Проверка оценки сложности.***

Временная сложность будет зависеть от количества перемещений дисков, которое равно

О(2^(n-1)) (т.к. при увеличении размера задачи, количество ходов увеличивается в 2 раза), где n – количество дисков, которые нужно переместить на другой стержень. Эта сложность приблизительно равна О(2^n)

|  |  |  |  |
| --- | --- | --- | --- |
| **Кол-во дисков** | **Количество перемещений** | **2^n** | **c** |
| 3 | 7 | 8 | 1,142 |
| 4 | 15 | 16 | 1.067 |
| 5 | 31 | 32 | 1.032 |
| 6 | 63 | 64 | 1.016 |

Т.к. отношение 2^n к количеству обращений к функции постоянно во всех 3 случаях, то предположение о сложности функции верно.