ГУАП

КАФЕДРА № 52

ПРЕПОДАВАТЕЛЬ

|  |  |  |
| --- | --- | --- |
| Доцент, кандидат |  | Е. М. Линский |
| технических наук |  |  |
|  |  |  |
| Должность, уч. степень, | подпись, дата | инициалы, фамилия |
| звание |  |  |
|  |  |  |

ОТЧЕТ О КУРСОВОЙ РАБОТЕ

''Пентамино''

по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |
| --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 5023 |  | В. А. Сильченко |
|  |  |  |  |
|  |  | подпись, дата | инициалы, фамилия |
|  |  |  |  |

Санкт-Петербург 2021

**Содержание**

|  |  |  |  |
| --- | --- | --- | --- |
| [1.](#page3) | [Постановка задачи.........................................................................................................................](#page3) | | [3](#page3) |
| [2.](#page3) | [Теория............................................................................................................................................](#page3) | | [3](#page3) |
| [3.](#page4) | [Алгоритм .......................................................................................................................................](#page4) | | [4](#page4) |
|  | [3.1](#page4) | [Идея алгоритма укладки (backtrack): .....................................................................................](#page4) | [4](#page4) |
|  | [3.2](#page4) | [Описание................................................................................................................................](#page4) | [4](#page4) |
|  | [3.3](#page5) | [Реализация алгоритма............................................................................................................](#page5) | [5](#page5) |
|  | [3.4](#page6) | [Анализ сложности .................................................................................................................](#page6) | [6](#page6) |
| [4.](#page7) | [Инструкция пользователя .............................................................................................................](#page7) | | [7](#page7) |
| [5.](#page8) | [Тестовые примеры ........................................................................................................................](#page8) | | [8](#page8) |
| [6.](#page9) | [Список литературы .......................................................................................................................](#page9) | | [9](#page9) |

1. **Постановка задачи**

Задачей данной курсовой работы является реализовать алгоритм укладки пентамино без наложения на языке С++.

1. **Теория**

Существует 12 видов фигур пентамино, которые можно обозначить прописными латинскими буквами. (Рис.1).

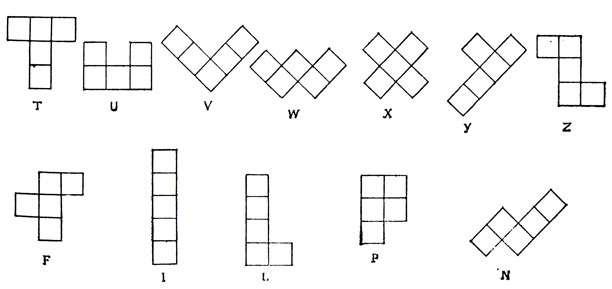


Рис.1: 12 фигур пентамино

* Каждая фигура занимает 5 клеток. 12 фигур пентамино
* Фигуры можно поворачивать на 90°.
* Поскольку всего имеется 12 разных пентамино и каждая из этих фигур покрывает пять клеток доски, то вместе они покрывают 60 клеток.
* Все возможные поля для укладки: 3x20, 4x15, 5x12, 6x10.

1. **Алгоритм**

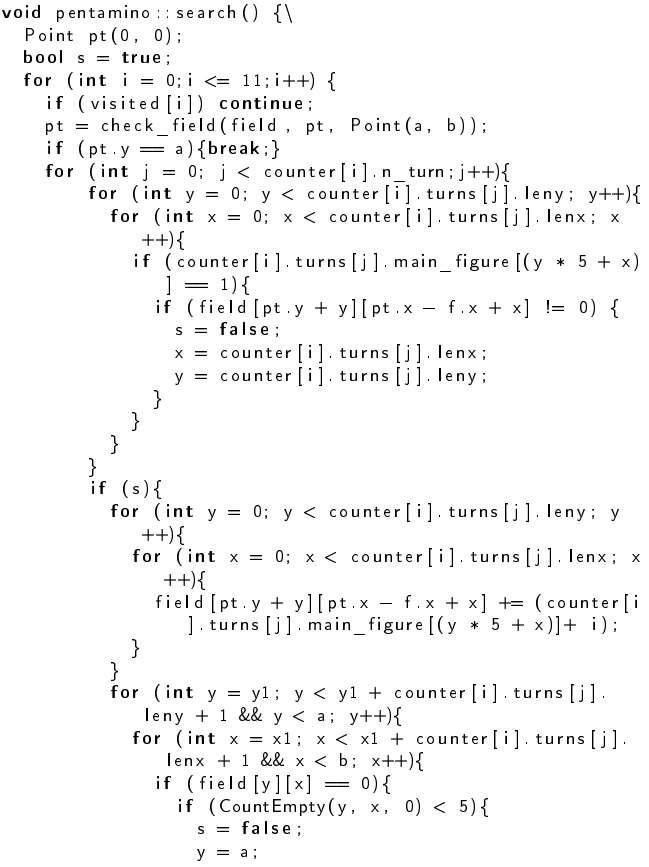
**3.1 Идея алгоритма укладки (backtrack):**

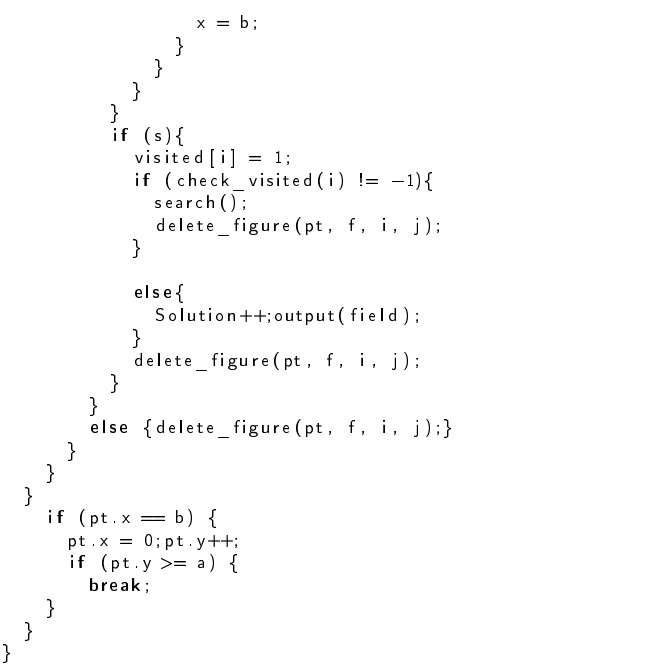
Алгоритм поиска с возвращением заключается в следующем. Строится конструктивно последовательное расширение частичного решения. Если на конкретном шаге такое расширение провести не удается, то происходит возврат к предыдущему частичному решению, и попытки его расширить продолжаются.

**3.2 Описание**

Поиск решения происходит в рекурсивной функции (Search). В функции Search (в цикле) определяется (с помощью функции checkfield) номер ячейки, которая ещё не занята пентамино и дальше в цикле происходит попытка расположить заданную пентамино в заданную позицию с учётом всех возможных вариантов поворота пентамино, если не удалось этого сделать, то переходим к другой фигуре. Если удалось расположить пентамино, то происходит анализ на наличие клеток, которые уже никак не удастся заполнить (менее четырёх клеток, которые изолированы в исходной фигуре при текущем расположении пентамино), такой анализ осуществляется с помощью функции CountEmpty для всех клеток в окружении располагаемого сейчас пентамино. Если «зажатых» клеток не обнаружено, то текущее расположение пентамино запоминается в массиве visited и происходит переход к расположению следующей фигурки пентамино (происходит вызов функции Search). При обнаружении «зажатых» клеток новое расположение пентамино удаляется с поля и происходит переход на анализ расположения пентамино в следующую пустую клетку поля. Если все клетки были заполнены, то происходит запись результата в выходной файл «output.txt» и анализ продолжается. Когда все варианты были рассмотрены, происходит выход из функции Search, после чего программа завершается, закрывая файл "output.txt".

**3.3 Реализация алгоритма**





* pt - структура, которая является текущим ходом;
* field - поле заполнения;
* deletefigure - функция удаления фигуры с поля
* CountEmpty - функция проверка изолированных клеток

**3.4 Анализ сложности**

Сложность алгоритма O((n\*m)!), где n - количество фигур, m - количество их зеркальных отражений и поворотов.

* 1. **Инструкция пользователя**

C помощью командной строки пользователь вводит прямоугольное поле размерностью в 60 клеток, с минимальной высотой 3. Открывается исполняемый файл. (Рис.2).

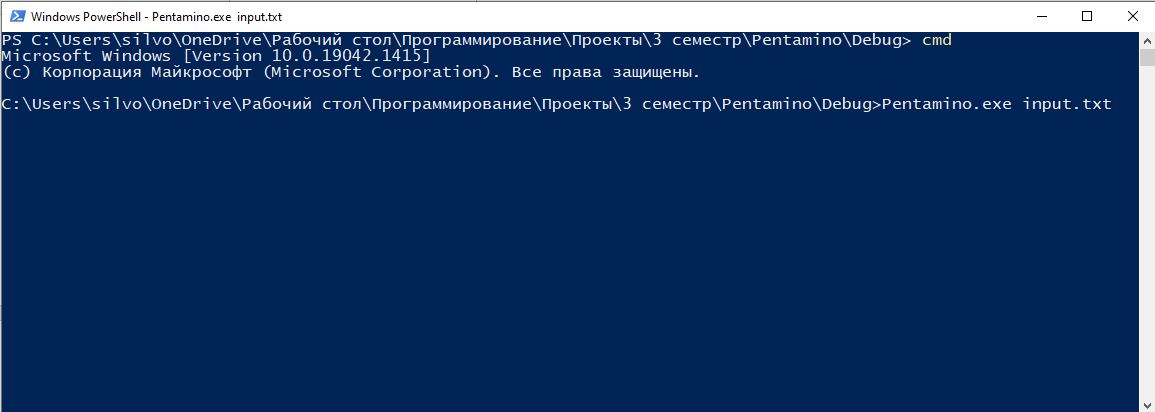


Рис.2: Запуск программы через консоль

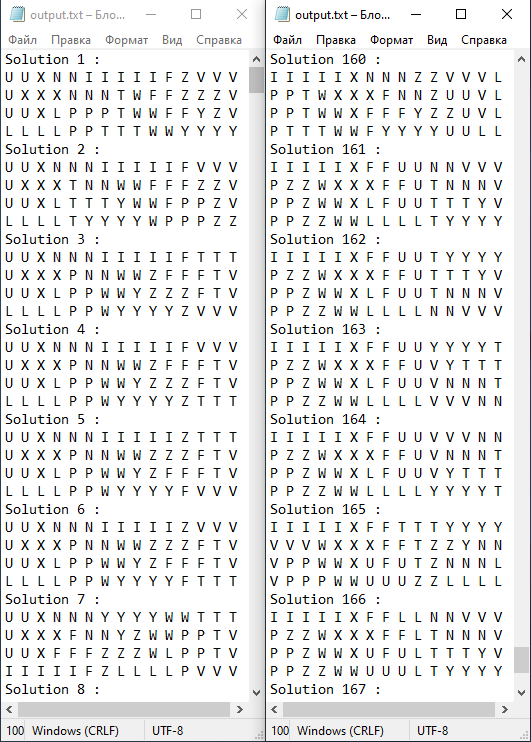
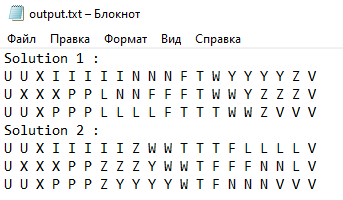
После запуска программа создаст файл в папке с программой "output.txt", в котором будут выведены все возможные комбинации заполнения заданного поля. (Рис.3).

Рис.3: Выходной файл

1. **Тестовые примеры**

Пользователь запускает программу с полем 3 на 20; 5 на 12.(Рис.4).



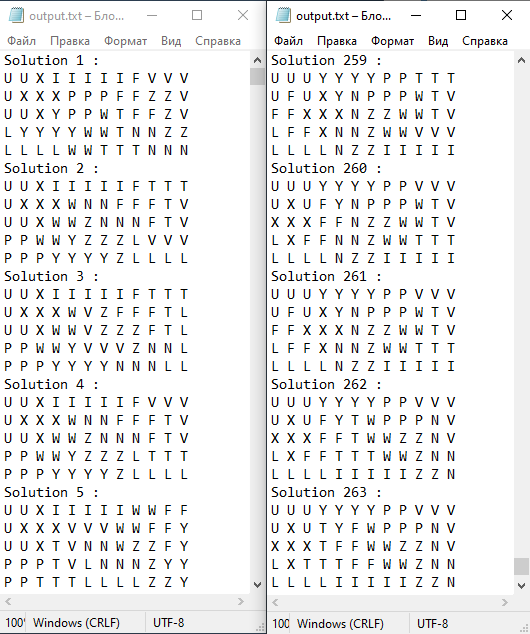


Рис.4: Вывод программы

* 1. **Список литературы**

1. **Рейнгольд Э., Нивергельт Ю., Део Н.** *Комбинаторные алгоритмы.* б.м. : М. Мир, 1980. (упражнения к гл. 4).
2. **Голомб⠀С.** *Пентамино.* б.м. : Мир, 1975.