|  |  |
| --- | --- |
|  | **Министерство образования и науки Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Робототехники и комплексной автоматизации

КАФЕДРА Системы автоматизированного проектирования (РК-6)

**ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

Студент Сидоров Михаил Михайлович

Группа РК6-56Б

Тип задания Лабораторная работа

Тема лабораторной работы Циклический сдвиг перестановки

Вариант 20

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сидоров М.М.**

*подпись, дата фамилия, и.о.*

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Волосатова Т.М.**

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Москва, 2024 г.*

**Оглавление**

[Задание на лабораторную работу](#_gjdgxs) 2

[Идея решения](#_3znysh7) 2

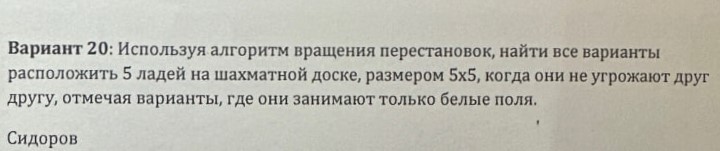
[Алгоритм](#_aplwgwmhwdq3) 2

[Листинг программы](#_5sw567tz43a8) 4

[Результаты работы программы.](#_v48ozo3o72q6) 6

[Источники](#_77pwe7h8fld2) 7

# Задание на лабораторную работу



# Идея решения

Основная идея решения заключается в использовании алгоритма перестановок для генерации всех возможных расположений ладей на шахматной доске размером 5x5, при которых они не угрожают друг другу. Каждая перестановка проверяется на соответствие следующим условиям:

1. Ладьи не находятся на одной линии (строке или столбце).
2. Ладьи занимают только белые поля.

Для проверки белых полей используется формула: если сумма индексов строки и столбца клетки чётная, клетка является белой.

# Алгоритм

1. **Инициализация**:

* Создается начальная конфигурация, представленная вектором индексов curr\_vec, где каждый элемент определяет положение ладьи в строке и столбце на доске.

1. **Функции**:

* **ShiftPerm**: выполняет циклический сдвиг (вращение) элементов вектора, начиная с указанной позиции, генерируя таким образом новую перестановку.

Алгоритм вращения основан на организации циклического сдвига элементов перестановки. Естественный способ перечисления перестановок циклическим сдвигом состоит в том, что начав с некоторой исходной перестановки, нужно последовательно сдвигать по циклу на 1 позицию влево все n элементов. При каждом сдвиге 1-й элемент текущей перестановки перемещается на последнее место без изменения взаимного расположения остальных элементов, образуя новую перестановку. Такая организация цикла сдвига называется вращением. Вращение всех элементов следует продолжать пока оно порождает оригинальные перестановки, не встречавшиеся ранее. Перестановка считается оригинальной, когда после сдвига позиция последнего элемента вращаемой части не равна его позиции в исходной перестановке. Когда в результате очередного вращения образуется ранее порожденная перестановка, нужно исследовать возможность построить оригинальную перестановку, применяя вращение последовательно для первых j=n-1, n-2,...,2 элементов, при фиксированном положении (n-j) хвостовых элементов. Если локальное вращение 1<j<n элементов порождает оригинальную перестановку, следует продолжить вращение циклическим сдвигом всех n элементов. В противном случае (j=1), перебор считается завершенным, т. к. перечислены все n! перестановок. Рассмотренный процесс систематического перечисления перестановок циклическим сдвигом называется алгоритмом вращения.

* **CheckWhite**: проверяет, занимают ли ладьи только белые клетки. Для каждой клетки рассчитывается позиция и проверяется условие белого поля (curr\_vec[i] + i) % 2 == 0.
* **PrintResult**: выводит текущую конфигурацию расположения ладей на доске и отмечает, если она соответствует условию размещения только на белых полях.

1. **Основной цикл программы**:

* Запускается перебор всех перестановок с использованием функций ShiftPerm и CheckWhite. Если конфигурация удовлетворяет условиям, она выводится и учитывается в общем количестве решений.

# Листинг программы

#include <iostream>

#include <iostream>

#include <vector>

using namespace std;

void ShiftPerm(vector<int>& curr\_vec, int k)

{

int temp = curr\_vec[k];

for (int i = k; i > 0; i--)

{

curr\_vec[i] = curr\_vec[i - 1];

}

curr\_vec[0] = temp;

}

bool CheckWhite(vector<int>& curr\_vec, int &white\_counter)

{

for (int i = 0; i < 5; i++)

{

if ((curr\_vec[i] + i) % 2 != 0)

{

return false;

break;

}

if (i == 4)

{

white\_counter++;

return true;

}

}

}

void PrintResult(vector<int>& curr\_vec, bool check\_white\_result)

{

int col[5] = { 1, 2, 3, 4, 5 };

char row[5] = { 'a', 'b', 'c', 'd', 'e' };

for (int i = 0; i < 5; i++)

{

cout << row[i] << col[curr\_vec[i]] << " ";

}

if (check\_white\_result)

{

cout << "(all white)";

}

cout << endl;

}

int main()

{

int counter = 1;

int white\_counter = 0;

vector<int> curr\_vec = { 0, 1, 2, 3, 4 };

int j = 4;

int n = j;

PrintResult(curr\_vec, CheckWhite(curr\_vec, white\_counter));

while (j > 0)

{

ShiftPerm(curr\_vec, j);

if (curr\_vec[j] != j)

{

PrintResult(curr\_vec, CheckWhite(curr\_vec, white\_counter));

counter++;

j = n;

}

else

{

j--;

}

}

cout << counter << endl;

cout << white\_counter;

}

# Результаты работы программы.

Программа нашла все возможные перестановки расположения 5 ладей на доске 5x5, которые не угрожают друг другу. Ниже приведены несколько примеров:

1. (1, 1) (2, 3) (3, 5) (4, 2) (5, 4) (all white)
2. (1, 3) (2, 1) (3, 4) (4, 5) (5, 2)
3. (1, 5) (2, 2) (3, 1) (4, 4) (5, 3)

Общее количество уникальных перестановок: **120**  
Количество перестановок, где ладьи занимают только белые поля: **12**

# Источники

http://bigor.bmstu.ru/?cnt/?prn=y/?doc=PO\_CAD/9.1.aux