ФГБОУ ВО “Чувашский государственный университет им.

И.Н. Ульянова”

Кафедра вычислительной техники

Лабораторная работа №3

**ИСЧЕРПЫВАЮЩИЙ ПОИСК**

Выполнил:

Студент гр. ИВТ-41-22

Ярдыков Эдуард Е.

Проверил:

кандидат технических наук

Павлов Леонид Александрович

Чебоксары 2023

***Цель работы*** – ознакомление с методами решения комбинаторных задач, получение навыков программирования оптимизационных задач.

**Подготовка к работе**

1. Разработать алгоритм и программу решения задачи коммивояжера с использованием общей схемы решения методом ветвей и границ [ ].

определить

// последнее сохраненное решение есть

// решение с наименьшей стоимостью

3. Разработать алгоритм и программу приближенного решения задачи коммивояжера.

}

}

}

}

}

}

Индивидуальные задания

5. Определить все возможные маршруты коня, начинающиеся на одном заданном поле шахматной доски и оканчивающиеся на другом. Никакое поле не должно встречаться в одном маршруте дважды.

int main() {

setlocale(0, "");

printf("Введите две пары чисел: {начY,начX} {конY,конX}: \n");

int nachY, nachX, konY, konX;

nachY = 3; nachX = 3; konY = 0; konX = 0;

vector<vector<int>> DOSKA(4, { 0,0,0,0 });

DOSKA[nachY][nachX] = 1;

DOSKA[konY][konX] = 2;

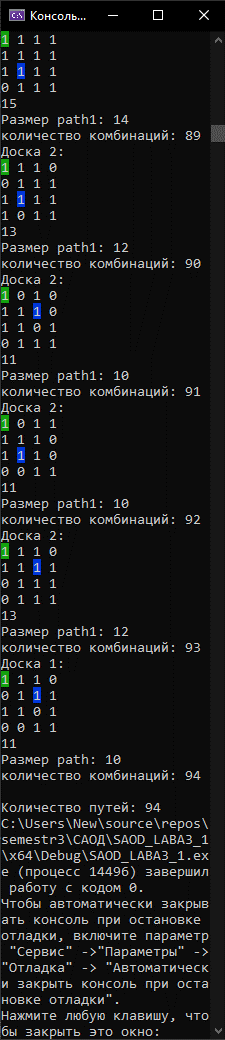
print\_doska(DOSKA);

cout <<endl;

doska\_resh(DOSKA, nachY, nachX, konY, konX);

printf("\nКоличество путей: %zd", all\_path.size());

}



5. Найти все простые несократимые дроби, заключенные между 0 и 1, знаменатели которых не превышают 9 (дробь задается двумя натуральными числами – числителем и знаменателем).

#include <iostream>

int get\_gcd(int x1, int x2)

{

if (!x2) return x1;

return get\_gcd(x2, x1 % x2);

}

int main(int argc, char\*\* argv)

{

unsigned found = 0;

int N = 9;

for (int den = 2; den != N + 1; den++) // 1 < знаменатель < N + 1

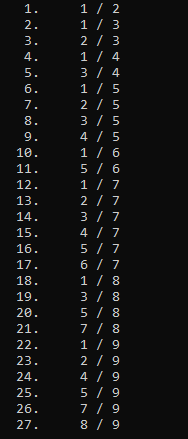
for (int nom = 1; nom != den; nom++) // 0 < числитель < знаменатель

if (get\_gcd(nom, den) == 1) // НОД

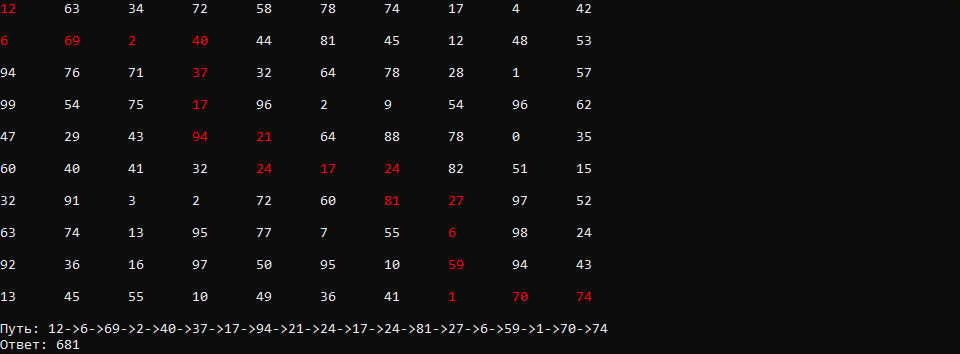
printf("%4u. %4d / %d\n", ++found, nom, den);

return 0;

}



6. Матрица размером *n* × *n* заполнена произвольными целыми числами (например, от 0 до 100). Необходимо найти такой путь из клетки (1, 1) до клетки (n, n), чтобы сумма чисел в клетках, через которые он пролегает, была минимальной. Нельзя перемещаться по диагонали.



Выводы: ознакомился с методами решения комбинаторных задач, получил навыки программирования оптимизационных задач.