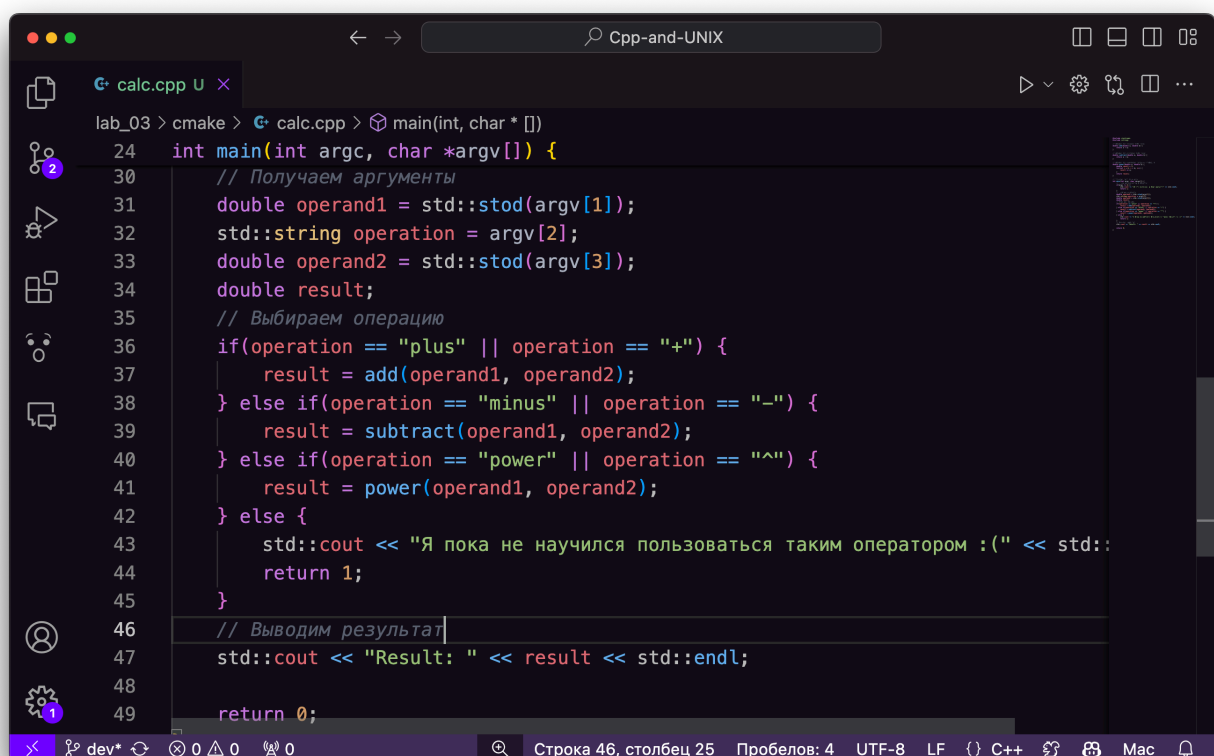
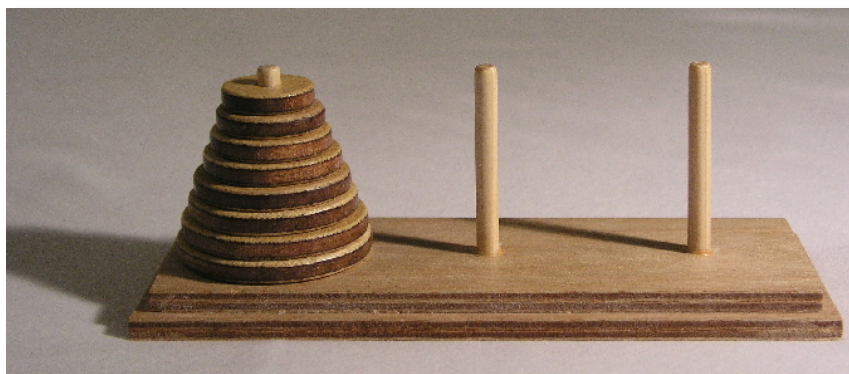


```
lab_03 > cmake > calc.cpp > main(int, char * [])
1  #include <iostream>
2  #include <string>
3
4  // Функция для сложения двух чисел
5  double add(double a, double b) {
6      return a + b;
7  }
8
9  // Функция для вычитания двух чисел
10 double subtract(double a, double b) {
11     return a - b;
12 }
13
14 // Функция для возведения числа a в степень b
15 double power(double a, double b) {
16     double result = 1;
17     for(int i = 0; i < b; i++) {
18         result *= a;
19     }
20     return result;
21 }
```



```
lab_03 > cmake > calc.cpp > main(int, char * [])
24 int main(int argc, char *argv[]) {
25     // Получаем аргументы
26     double operand1 = std::stod(argv[1]);
27     std::string operation = argv[2];
28     double operand2 = std::stod(argv[3]);
29     double result;
30     // Выбираем операцию
31     if(operation == "plus" || operation == "+") {
32         result = add(operand1, operand2);
33     } else if(operation == "minus" || operation == "-") {
34         result = subtract(operand1, operand2);
35     } else if(operation == "power" || operation == "^") {
36         result = power(operand1, operand2);
37     } else {
38         std::cout << "Я пока не научился пользоваться таким оператором :(" << std::endl;
39         return 1;
40     }
41     // Выводим результат
42     std::cout << "Result: " << result << std::endl;
43     return 0;
44 }
```

2. [C++ RECURSION] Решить задачу ханойской башни с использованием рекурсии

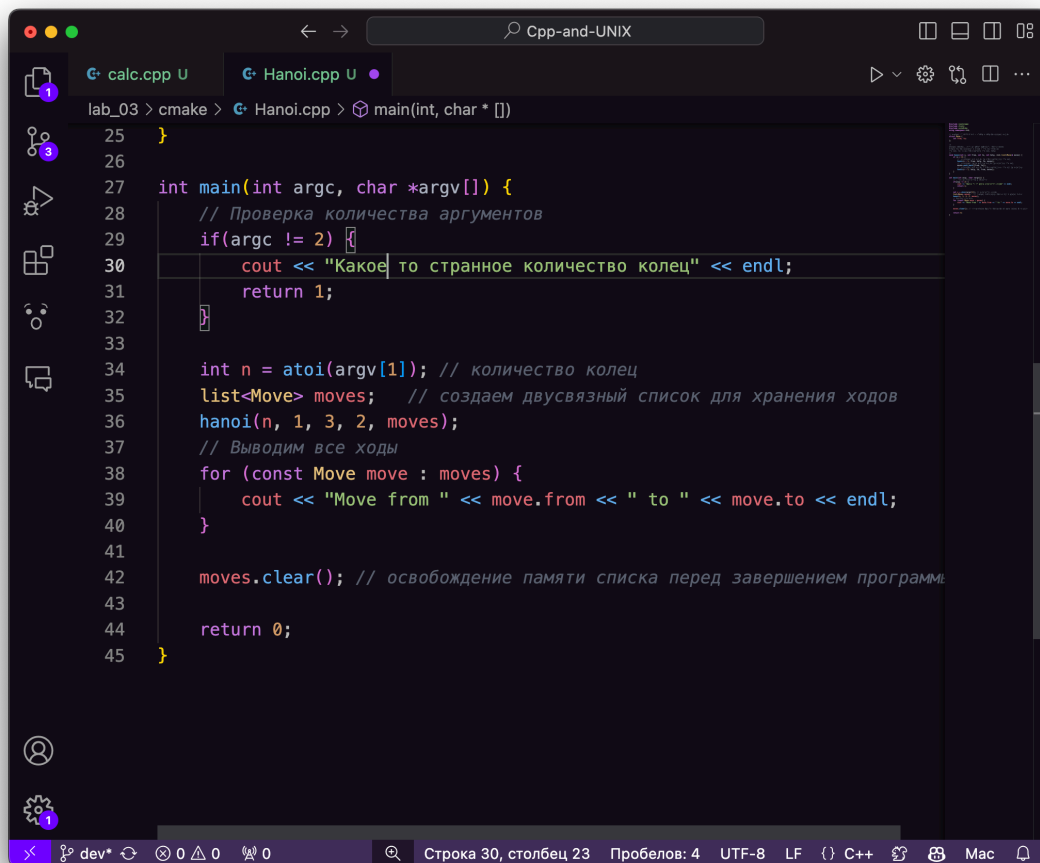


Описание: Ханойская башня является одной из популярных головоломок XIX века. Даны три стержня, на один из которых нанизаны восемь колец, причём кольца отличаются размером и лежат меньшее на большем. Задача состоит в том, чтобы перенести пирамиду из восьми колец за наименьшее число ходов на другой стержень. За один раз разрешается переносить только одно кольцо, причём нельзя класть большее кольцо на меньшее.

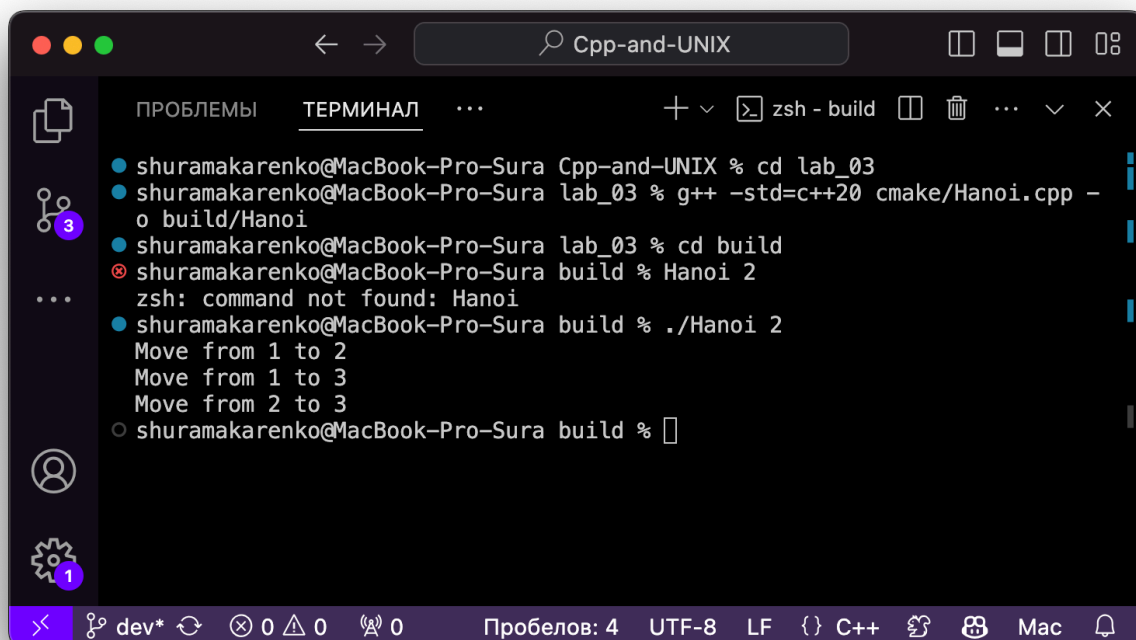
Результат обнаруженной последовательности шагов записать в виде двусвязного списка. В конце программы сделать вывод этого списка на экран. Освободить память списка перед завершением программы

Решение

```
lab_03 > cmake > C Hanoi.cpp > main(int, char * [])
1  #include <iostream>
2  #include <list>
3  #include <cstdlib>
4  using namespace std;
5
6  // Создаем структуру ход – откуда и куда перемещаем кольцо
7  struct Move {
8      int from, to;
9  };
10
11 /*
12 Сделаем функцию, которая будет заполнять список moves
13 ходами для перемещения n колец с стержня from на
14 стержень to через вспомогательный стержень help
15 */
16 void hanoi(int n, int from, int to, int help, std::list<Move>& moves) {
17     if (n > 0) {
18         // Перемещаем n-1 кольцо на вспомогательный стержень
19         hanoi(n - 1, from, help, to, moves);
20         // Перемещаем самое большое кольцо на конечный стержень
21         moves.push_back({from, to});
22         // Перемещаем n-1 кольцо с вспомогательного стержня на конечный
23         hanoi(n - 1, help, to, from, moves);
24     }
25 }
26
27 int main(int argc, char *argv[]) {
28     // Проверка количества аргументов
29 }
```



```
lab_03 > cmake > G- Hanoi.cpp U main(int, char * [])
25 }
26
27 int main(int argc, char *argv[]) {
28     // Проверка количества аргументов
29     if(argc != 2) {
30         cout << "Какое то странное количество колец" << endl;
31         return 1;
32     }
33
34     int n = atoi(argv[1]); // количество колец
35     list<Move> moves; // создаем двусвязный список для хранения ходов
36     hanoi(n, 1, 3, 2, moves);
37     // Выводим все ходы
38     for (const Move move : moves) {
39         cout << "Move from " << move.from << " to " << move.to << endl;
40     }
41
42     moves.clear(); // освобождение памяти списка перед завершением программы
43
44     return 0;
45 }
```

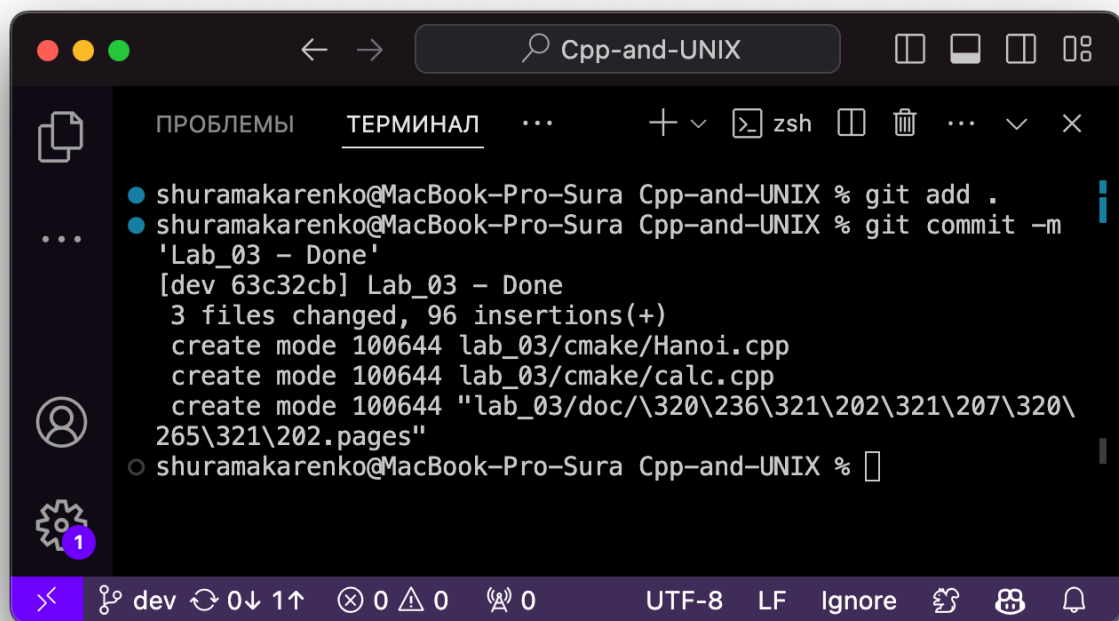


```
ПРОБЛЕМЫ ТЕРМИНАЛ ... + v zsh - build
shuramakarenko@MacBook-Pro-Sura Cpp-and-UNIX % cd lab_03
shuramakarenko@MacBook-Pro-Sura lab_03 % g++ -std=c++20 cmake/Hanoi.cpp -o build/Hanoi
shuramakarenko@MacBook-Pro-Sura lab_03 % cd build
shuramakarenko@MacBook-Pro-Sura build % Hanoi 2
zsh: command not found: Hanoi
shuramakarenko@MacBook-Pro-Sura build % ./Hanoi 2
Move from 1 to 2
Move from 1 to 3
Move from 2 to 3
shuramakarenko@MacBook-Pro-Sura build %
```

3. [SAVE] Результат всех вышеперечисленных шагов сохранить в репозиторий (+ отчет по данной ЛР в папку doc)

Фиксацию ревизий производить строго через ветку dev. С помощью скриптов накатить ревизии на stg и на prd. Скрипты разместить в корне репозитория. Также создать скрипты по возврату к виду текущей ревизии (даже если в папке имеются несохраненные изменения + новые файлы).

Решение



```
shuramakarenko@MacBook-Pro-Sura Cpp-and-UNIX % git add .
shuramakarenko@MacBook-Pro-Sura Cpp-and-UNIX % git commit -m 'Lab_03 - Done'
[dev 63c32cb] Lab_03 - Done
3 files changed, 96 insertions(+)
create mode 100644 lab_03/cmake/Hanoi.cpp
create mode 100644 lab_03/cmake/calc.cpp
create mode 100644 "lab_03/doc/\320\236\321\202\321\207\320\265\321\202.pages"
shuramakarenko@MacBook-Pro-Sura Cpp-and-UNIX %
```

Выводы

1. Научились писать калькулятор!!
2. Использовали рекурсию на Ханойских башнях