МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский университет «ИТМО» (НИУ ИТМО)

Лабораторная работа №3 «C++ CLI /FUNCTION / LOOP / RECURSION»

по курсу «C++ и UNIX системы»

Выполнил студент 3 курса группы К3333: Буданцев А.А.

Проверил: Маслов И.Д.

Санкт-Петербург - 2023

Оглавление

Цель работы

Познакомить студента с основными алгоритмическими конструкциями, которые будут использоваться для создания CLI программы. Далее продемонстрировать эффективность использования механизма рекурсии.

C++ алгоритмы: CLI Калькулятор вещественных чисел +, -, ^, . Реализация с использованием только функций, условий, циклов, + и -. Вид команд в консоли: calc plus / minus / power; Ханойская башня, результат корректной последовательности

Задача

1. **[C++ CLI CALC] Создать программу CALC с интерфейсом CLI**
   Создать программу под названием CALC, которая будет принимать на вход 3 аргумента (2 операнда и 1 оператор). Оператор может быть: +, -, ^. Реализация операторов только с использованием функций, условий, циклов, +, - и *.
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.

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

1. [C++ CLI CALC] Создать программу CALC с интерфейсом CLI

Основная функция программы, которая принимает аргументы в командной строке. Argc – количество аргументов, argv – массив строк (аргументов).

```
int main(int argc, char *argv[]){
    if (CheckArguments(argc, argv)) {
        double a = stod(argv[1]);
        double b = stod(argv[3]);
        char* s = argv[2];
        switch (*s) {
            case '+':
                cout << a << " + " << b << " = " << add(a,b) << "\n";
                break;
            case '-':
                cout << a << " - " << b << " = " << subtract(a,b) << "\n";
                break;
            case '^':
                cout << a << " ^ " << b << " = " << pow(a,b) << "\n";
                break;
            default:
                cout << "Wrong operator! " << "\n";
                break;
        }
    }

    return 0;
}
```

Реализация вспомогательных функций:

Проверка аргументов:

```
bool CheckNumbers(char *number) {
    for(char* p=&number[0]; *p != '\0'; p++) {
        if (isdigit(*p) == false)
            return false;
    }
    return true;
}

bool CheckArguments(int argc, char *argv[]){
    if (argc != 4) {
        cout << "Entered wrong count of arguments. Please enter 3 arguments!!!" << "\n";
        return false;
    }

    if (CheckNumbers(argv[1]) == false || CheckNumbers(argv[3]) == false) {
        cout << "Entered wrong type of arguments in 1 or 3 positions." << "\n";
        cout << "Please enter numbers in 1 and 3 positions!!!" << "\n";
    }

    return true;
}
```

Вычисления:

```
double add(double a, double b) {
    return a + b;
}

double subtract(double a, double b) {
    return a - b;
}

double pow(double a, double b) {
    int result = 1;
    for(int i=1; i<=b; i++)
        result *= a;
    return result;
}
```

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

   Реализация основной функции:

```
int main() {
    int N=8;

    Node* head= nullptr;

    MoveTower(N, 'A', 'C', 'B', &head);

    print(&head);

    while (head) {
        Node* temp = head;
        head = head->prev;
        delete temp;
    }

    return 0;
}
```

   Структура двухсвязного списка

```
struct Node {
    int Data;
    char from, to;
    Node* prev;
    Node* next;
};
```

Реализация функции с Ханойской башни через рекурсию:

```
void MoveTower(int disk, char point1, char point2, char temporaryPoint, Node** headPointer) {
    if (disk == 0)
        return;

    MoveTower(disk - 1, point1, temporaryPoint, point2, headPointer);

    Move(headPointer, disk, point1, point2);

    MoveTower(disk - 1, temporaryPoint, point2, point1, headPointer);
}
```

Функция Move для сохранения действий

```
void Move(Node** headPointer, int disk, char from, char to) {
    Node* newMove = new Node;
    newMove->Data = disk;
    newMove->from = from;
    newMove->to = to;
    newMove->prev = *headPointer;
    newMove->next = nullptr;

    if (*headPointer)
        (*headPointer)->next = newMove;

    *headPointer = newMove;
}
```

Вывод действий:

```
void print(Node** head) {
    Node* temp = *head;
    while (temp) {
        cout << "Move disk " << temp->Data << " from peg " << temp->from << " to peg " << temp->to << "\n";
        temp = temp->prev;
    }
}
```

3. **[SAVE] Результат всех вышеперечисленных шагов сохранить в репозиторий**

```
git checkout prd

git merge stg

git tag $(date '+%Y-%m-%d-%H-%M-%S')

git push --tags

git push

git checkout dev
```

```
git clean -dxf
git reset --hard
```

Вывод

В ходе работы изучена работа с CLI, указателями и применена рекурсия.