**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №1**

**по дисциплине «Алгоритмы и структуры данных»**

Тема: Рекурсия

|  |  |  |
| --- | --- | --- |
| Студент гр. 8381 |  | Гречко В.Д. |
| Преподаватель |  | Жангиров Т.Р. |

Санкт-Петербург

2019

## Цель работы.

Реализовать программу, использующую рекурсию, с целью ознакомления с основными понятиями и приёмами рекурсивного программирования и получения навыков программирования рекурсивных процедур и функций на языке программирования C++.

## Задание.

Напечатать все перестановки заданных *n* различных натуральных чисел (или символов).

## Основные теоретические положения.

В программировании рекурсия тесно связана с функциями, точнее именно благодаря функциям в программировании существует такое понятие как рекурсия или рекурсивная функция. Простыми словами, рекурсия – определение части функции (метода) через саму себя, то есть это функция, которая вызывает саму себя, непосредственно (в своём теле) или косвенно (через другую функцию).

Всё решение сводится к решению основного или, как ещё его называют, базового случая. Существует такое понятие как шаг рекурсии или рекурсивный вызов. В случае, когда рекурсивная функция вызывается для решения сложной задачи (не базового случая) выполняется некоторое количество рекурсивных вызовов или шагов, с целью сведения задачи к более простой. И так до тех пор пока не получится базовое решение.

Перестановка – это комбинация элементов из N разных элементов взятых в определенном порядке. В перестановке важен порядок следования элементов, и в перестановке должны быть задействованы все N элементов.

## Выполнение работы.

Написание работы производилось на базе операционной системы Linux Elementary OS в среде разработки CLion. Сборка, отладка и тестирование также производились в CLion.

Сначала программа получает на вход количество элементов в перестановке (n) и выделяет память под массив целочисленных переменных длины n. Далее происходит диалог с пользователем о выборе источника получения данных: предложенный файл text.txt или консольный ввод. В обоих случаях после считывания данных проверяется их корректность. В случае ошибки программа завершает работу и выводит соответствующее сообщение.

Во время считываниях данных из файла используется библиотека <fstream>.

Далее вызывается рекурсивная функция Permutation, которая принимает на вход исходный массив, его длину и счетчик. В случае, когда значения счетчика и длины совпадает (что означает полный проход по массиву и генерацию новой перестановки), то числа выводятся либо в консоль, либо в файл. Иначе запускается цикл с дополнительным счетчиком для генерации новой перестановки: происходит перестановка элементов i и j местами, а также рекурсивный вызов Permutation с увеличенным счетчиком. Также внутри этого цикла выполняется расчет глубины рекурсии.

Полученные результаты либо выводятся на консоль, либо записываются в файл.

## Оценка эффективности алгоритма.

Алгоритм, реализованный в программе, имеет линейную зависимость от размера массива, то есть сложность оценивается как O(n).

## Примеры работы программы.

|  |  |
| --- | --- |
| Входные данные | Результат |
| 1 4 7 3 | 1 4 7 3  1 4 3 7  1 7 4 3  1 7 3 4  1 3 7 4  1 3 4 7  4 1 7 3  4 1 3 7  4 7 1 3  4 7 3 1  4 3 7 1  4 3 1 7  7 4 1 3  7 4 3 1  7 1 4 3  7 1 3 4  7 3 1 4  7 3 4 1  3 4 7 1  3 4 1 7  3 7 4 1  3 7 1 4  3 1 7 4  3 1 4 7 |
| 12 34 101 | 12 34 101  12 101 34  34 12 101  34 101 12  101 34 12  101 12 34 |

## Выводы.

В ходе выполнения лабораторной работы была реализована программа, находящая перестановки рекурсивным методом и выводящая их пользователю. Были изучены основные понятия и приёмамы рекурсивного программирования

# Приложение А Исходный код программы

файл main.cpp:

#include <iostream>

#include <string>

#include <fstream>

#include <fstream>

using namespace std;

char choice;

int depht = 0;

void Permutation(int \*str,int i,int len){

if(choice == 'f'){

ofstream fil\_n;

fil\_n.open("/home/veronica/programming/labfirst/text.txt", ios::app); //opening file to add results

if ( i == len ){

for(int j = 0; j < len; j++){

fil\_n << str[j] << ' ';

}

fil\_n << endl;

fil\_n.close();

}

else{

for(int j = i; j < len; j++){

swap(str[i], str[j]);

depht++;

Permutation(str, i + 1, len);

swap(str[i], str[j]);

}

}

}

else{

if ( i == len ){

for(int j = 0; j < len; j++){

cout << str[j] << ' ';

}

cout << endl;

}

else{

for(int j = i; j < len; j++){

swap(str[i], str[j]);

depht++;

Permutation(str, i + 1, len); // cout temporary results

swap(str[i], str[j]);

}

}

}

}

int main() {

int \*mass;

int n;

cout << "Введите количество чисел:" << endl; // make check what all numbers are integer

cin >> n;

mass = new int[n];

cout << "Откуда будут взяты значения?";

cin >> choice;

if (choice == 'c'){

cout << "Введите числа:" << endl;

for(int i =0; i< n; i++){

cin >> mass[i];

}

if(cin.fail()){

cout << "You have entered wrong input" <<endl;

return 0;

}

Permutation(mass, 0, n);

}

else {

ifstream fil;

fil.open("/home/veronica/programming/labfirst/text.txt"); // open file to get information

for(int i =0; i< n; i++) {

fil >> mass[i];

if(fil.fail()){

cout << "You have entered wrong input" <<endl;

return 0;

}

}

if(fil.fail()){

cout << "You have entered wrong input" <<endl;

return 0;

}

if(!fil.is\_open()) cout<<"ERROR";

fil.close();

Permutation(mass, 0, n);

}

cout << depht <<endl;

return 0;

}