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. 9304 _____ Атаманов С.Д.

Преподаватель _____ Филатов Ар. Ю.

Санкт-Петербург

2020

Цель работы.

Изучить основы работы с рекурсией на примере языка C++.

Задание.

Вариант 1.

Для заданных неотрицательных чисел n и m вычислить (рекурсивно) биноминальные коэффициенты, пользуясь их определением:

$$C_n^m = \begin{cases} 1, если\ m = 0, n > 0\ или\ m = n \geq 0, \\ 0, если\ m > n \geq 0, \\ C_{n-1}^{m-1} + C_{n-1}^{m}\ в\ остальных\ случаях \end{cases}$$

Выполнение работы.

Для работы алгоритма был создан класс Combination, который имеет 3 публичных поля: переменная типа unsigned long long int C, которое хранит в себе сам биноминальный коэффициент, по умолчанию инициализировано нулем; переменные типа int n и m, которые хранят в себе числа m, и n соответственно. Также в классе реализована рекурсивная функция unsigned long long int combination(int n, int m), которая выполняет рекурсивный подсчет биноминального коэффициента, исходя из его определения: функция возвращает 1, если m=0 и n>0 или если m=n>=0; 0, если m>n>=0; и сумму вызова двух функций combination() с переменными m-1, n-1 и m, n-1.

В функции main() производится чтение аргументов командной строки, присваивание полям m и n класса Combination значений которые были переданы программе. Если количество аргументов не равно двух, выводится сообщение о нехватке аргументов и выполнение программы прекращается. Затем производится вызов функции combination() и вывод значения биноминального коэффициента в stdout.

Сама программа получает на вход два числа, переданные как аргументы командной строки. Результат программа выводит в стандартный поток вывода.

Разработанный программный код смотри в приложении А.

Тестирование.

Для тестирования был написан bash-скрипт, который создает файл для записи выходных данных, затем передает программе данные для тестирования, сравнивает выходные данные с истинным данными, которые записаны в файле с ответами и, в зависимости от ответа, приписывает к выходным данным лейбл “Correct”, если ответ верный и “Incorrect”, если ответ неверный.

Результаты тестирования смотри в приложении Б.

Выводы.

В ходе выполнения работы были изучены основы работы с рекурсивными алгоритмами и разработана программа на языке C++, которая рекурсивно вычисляет биноминальные коэффициенты.

ПРИЛОЖЕНИЕ А.

ИСХОДНЫЙ КОД ПРОГРАММЫ.

Название файла: main.cpp

```
#include <iostream>
class Combination{

public:
	unsigned long long int C = 0;
	int n;
	int m;
	unsigned long long int combination(int n, int m){
   if((m == 0 && n>0) || (m == n && n>=0))
		return 1;
	else if(m > n && n >=0)
		return 0;
	else
		return combination(n-1, m-1) + combination(n-1, m);
   }
};

int main(int argc, char* argv[]){
	Combination newton;
	if(argc!=3){
		std::cout << "Don't have enough argument's\n";
		return 0;
	}
	newton.n = atoi(argv[1]);
	newton.m = atoi(argv[2]);

	newton.C = newton.combination(newton.n, newton.m);
	std::cout << newton.C << "\n";
	return 0;
}#include <iostream>
class Combination{

public:
	unsigned long long int C = 0;
	int n;
	int m;
	unsigned long long int combination(int n, int m){
   if((m == 0 && n>0) || (m == n && n>=0))
		return 1;
	else if(m > n && n >=0)
		return 0;
	else
		return combination(n-1, m-1) + combination(n-1, m);
   }
};
```

```
int main(int argc, char* argv[]){
	Combination newton;
	if(argc!=3){
		std::cout << "Don't have enough argument's\n";
		return 0;
	}
	newton.n = atoi(argv[1]);
	newton.m = atoi(argv[2]);

	newton.C = newton.combination(newton.n, newton.m);
	std::cout << newton.C << "\n";
	return 0;
}
```

ПРИЛОЖЕНИЕ Б.

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ.

Входные данные	Выходные данные	Комментарий
1	Don’t have enough argument’s	Вывод для проверки поведения программы при неправильных значениях
0 0	1	Проверка ввода одинаковых значений m и n
0 1	0	Проверка ввода при m>n
1 0	1	Простейший тест
5 4	5	Вычисление коэффициента
10 3	120	Вычисление коэффициента
13 4	715	Вычисление коэффициента
12 13	0	Проверка ввода при m>n
30 15	541931236	Вычисление больших значений
100 93	155117520	Вычисление больших значений
100 93	16007560800	Вычисление большого самого большого значения
100 500	0	Проверка большого значения при m>n