

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

Преподаватель

Прокофьев М.Д.

Филатов А.Ю.

Санкт-Петербург

Цель работы.

Узнать о рекурсии и о ее использовании в практике

Задание.

Функция $f(n)$ определена для целых положительных чисел:

$$f(n) = \begin{cases} 1, & \text{если } n = 1 \\ \sum_{i=2}^n f(n \operatorname{div} i), & \text{если } n \geq 2 \end{cases}$$

Вычислить $f(k)$ для $k=15,16,\dots,30$

Выполнение работы.

Для выполнения работы была создана функция: $f(\text{int } n)$. Рекурсивная функция $f(n)$ сделана в соответствии с условиями задачи. При ее вызове с определенным аргументом суммируются последовательно функции, аргументы которых равны целому числу от деления аргумента на, соответственно, итератор. Если аргумент функции равен 1, то и сама функция равна 1. В угоду меньшего заполнения стека в самой функции не вызывается рекурсия для $f(2)$ или $f(1)$, они “заведомо в программе равны” 1. Поэтому существует переменная “quantity” которая отвечает за количество “единиц”. Всего в сумме слагаемых $(n-1)$, учитывая то что складывается суммы с итерированием от 2 до n . По некоторой закономерности, количество слагаемых, которые не равны 1, и которые соответственно “нуждаются в вызове рекурсией”, равно $(n/3)-1$. Соответственно, переменная “quantity” равна $((n-1)-((n/3)-1)=n-(n/3))$ Таким образом, сначала вычитывается сумма тех функций, где рекурсия просто необходима, а потом прибавляется определенное количество единиц к сумме. Соответственно уменьшается количество данных, идущих в стек.

Функция `main` принимает строку `argv[]`, которая позже преобразуется в `int` с помощью функции `atoi` из библиотеки `stdlib` и запишется в переменную `res`, от которой будет вызываться функция `f`. Посредством `testing.py`, в `argv[]` последовательно передаются строки из файлов `SuccessTests.txt` и `ErrorTests.txt` для тестирования.

Выводы.

Изучена рекурсия. Написана рекурсивная функция, включающая в себе сумму рекурсивных функций. Использование рекурсии в решении этой задачи является оправданным поскольку, как минимум, функция, заданная в условии, вызывает саму себя.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Файл: main.cpp

```
#include <iostream>
```

```
#include <stdlib.h>
```

```
int f(int n)
```

```
{
```

```
    int sum=0, i=1, quantity=n-(n/3);
```

```
    if((n==2)||(n==1)) return 1;
```

```
    while((n/++i)>2) sum+=f(n/i);
```

```
    sum+=quantity;
```

```
    return sum;
```

```
}
```

```
int main (int argc, char* argv[])
```

```
{
```

```
    int res=0;
```

```
    if(res<1) std::cout << "not answer";
```

```
    else std::cout << f(res);
```

```
    s
```

```
    =
```

```
    a
```

```
    t
```

```
    o
```

```
    i
```

```
    (
```

```
);
```

```
r
```

```
e
```

```
t
```

ПРИЛОЖЕНИЕ В

ТЕСТИРОВАНИЕ

Результаты тестирования представлены в таблице Б.1

Таблица Б.1 — Результаты тестирования

№ п/п	Входные данные	Выходные данные
1.	15	25
2.	b5	25
3.	-14b	not answer
4.	14b-	22
5.	150000	196854853
6.	-	not answer
7.	1	1
8.	0	not answer
9.	-1	not answer
10.	30	84