

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.

Фиалковский М.С.

Санкт-Петербург

2020

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 были созданы две функции: $f(\text{int } n)$ и $\text{TextToInt}(\text{char } *c)$

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.

Также в программе присутствует функция $\text{TextToInt}(c)$. Эта функция служит для преобразования ввода(который может в виде текста) в число. В самой функции определено число, которое строится по введенной строке c . Если в строке присутствуют лишние элементы, по типу букв, запятых или иных ASCII

символов кроме цифр и минуса, они удаляются из строки. (Например, был ввод: “15g56hf3”, он преобразовался в “15563”)

Кроме того, в главной функции учтен ввод числа меньшего 1, в таком случае пользователю предлагается заново ввести число.

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>

using namespace std;

int TextToInt(char *c)
{
    bool warning=0;
    int znak=1;
    int result=0;
    for(int i=0; i<(strlen(c)); i++)
    {
        if((c[i] == '-')&&(znak==1)&&(result==0)) znak*=(-1);
        if((c[i] >= '0') && (c[i] <= '9'))
        {
            result=result*10+c[i]-'0';
        } else if(c[i] != '-') warning=1;
    }
    result*=znak;
    return result;
}

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()
{
    char vvod[256]; int res=0;
    gets(vvod);
    res=TextToInt(vvod);
    if(res<1) cout << "not answer";
    cout << f(res);
    return 0;
}
```

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.	15b5	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