

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МОЭВМ**

**отчет**  
**по лабораторной работе №1**  
**по дисциплине «Алгоритмы и структуры данных»**  
**Тема: Рекурсия**

Студент гр.8304

\_\_\_\_\_

Холковский К.В.

Преподаватель

\_\_\_\_\_

Фирсов М.А.

Санкт-Петербург

2019

### **Задание.**

#### **5-й Вариант**

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  $k = 15, 16, \dots, 30$ .

### **Цель работы.**

Ознакомиться с основными понятиями и приёмами рекурсивного программирования, получить навыки программирования рекурсивных функций на языке программирования C.

### **Описание алгоритма.**

После получения исходного числа  $n$ , оно проверяется на равенство с 1, если это так, то функция возвращает единицу, в противном случае запускается цикл от  $i=2$  до исходного  $n$  и в некоторую переменную  $sum$  записывается сумма результатов вызова функцией себя с параметром  $n \operatorname{div} i$ , после чего возвращается значение переменной  $sum$ .

Для отслеживания промежуточных результатов и глубины рекурсии была создана глобальная переменная  $counter$ , в которой хранится глубина рекурсии, и функция `void test(int k)`, принимающая параметр с которым вызывается в данный момент функция `int f(int)` и печатающая некоторое кол-во отступов и значение параметра.

### **Описание функций.**

`int f(int n);` - Функция выполняет следующий алгоритм:

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

### **Выводы.**

Для решения полученной задачи целесообразно было использовать рекурсию.

## Тестирование

n	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
f(n)	25	29	30	36	37	42	45	47	48	60	62	64	68	73	74	84

```
Введите n: 7
f(7)
    f(3)
        f(1)
        f(1)
    f(2)
        f(1)
    f(1)
    f(1)
    f(1)
    f(1)
f(7)=7
```

Вывод с промежуточными результатами для n=7

n	1	0	-4
f(n)	1	0 is Incorrect n	-4 is Incorrect n

## Исходный код

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

static int counter=0;//для подсчета глубины рекурсии
static int flag=0;//для (не)вывода промежуточных значений

void test(int k){//для вывода глубины рекурсии и промежуточных значений
    for(int i=0;i<counter;i++)
        printf("        ");
    printf("f(%d)\n",k);
}

int f(int n){//функция из условия
    int sum=0;
    if (n==1){
        counter--;
```

```

        return 1;
    }

    else{
for(int i=2;i<=n;i++){
        if(flag)
            test(n/i);

        counter++;
        sum+=f(n/i);
    }
    counter--;
}
return sum;
}

int main(int argc,char* argv[]){
    int n,k;
    if(argc==1){
        printf("Введите n: ");
        if((scanf("%d",&n)==0)|| (n<1)){
            printf("Incorrect n\n");
            return 0;
        }

        flag=1;
        test(n);
        counter++;
        printf("f(%d)=%d\n",n,f(n));
    }
    else{
        char* path=(char*)calloc(strlen(argv[1])+7,sizeof(char)); //открыть файл
        strcat(path,"Tests/");
        strcat(path,argv[1]);
        FILE* file=fopen(path,"r");
        if(!file){
            printf("Incorrect file name\n");
            return 0;
        }

        k=fscanf(file,"%d\n",&n);
        char * str=(char*)calloc(100,sizeof(char));
        while(k!=EOF){
            if(k==1)
                if(n>0)
                    printf("f(%d)=%d\n\n",n,f(n));
                else

```

```
        printf("%d is Incorrect n\n\n",n);
    else{
        fgets(str,99,file);
        printf("%s^^^ is Invalid format n\n\n",str);
    }
    k=fscanf(file,"%d\n",&n);
}
free(path);
free(str);
}
return 0;
}
```