

**Федеральное государственное бюджетное образовательное учреждение
высшего образования
"Уфимский государственный авиационный технический университет"**

Кафедра Высокопроизводительных вычислительных технологий и систем

Дисциплина: Программирование

Отчет по лабораторной работе № 2

Тема: Функции языка Си

Группа ПМ-153	Фамилия И.О.	Подпись	Дата	Оценка
Студент	Шамаев И.Р.			
Принял	Гайнетдинова А.А.			

Уфа 2019

Цель: ознакомиться с принципами разбиения программ на блоки и правилами оформления функций на языке Си.

Теоретический материал

Были использованы операторы if-else (условные), for (цикла). Применены макрос-функция (с помощью #define) и функции, написанные в отдельной библиотеке, прототипы которых находятся в заголовочном файле (подключается через #include “имя.h”) и в файле с исходным кодом.

Ответы на контрольные вопросы

1. Приведите структуру (формат) определения функции.

Определение функции состоит из двух частей: её заголовка и блок (тело функции):

```
тип_возвращаемого_значения имя_функции(список_параметров)
{
    операторы
    return n;
}
```

2. Приведите структуру (формат) вызова функции.

```
имя_функции(аргумент1, аргумент2,...);
```

3. С помощью какого оператора производится возврат из функции в точку ее вызова?

Оператор return.

4. Что такое прототип (объявление) функции?

Это объявление функции, не содержащее тела функции, но указывающее имя функции, типы аргументов и возвращаемый тип данных.

5. Где размещают прототипы функций?

Прототип функции должен быть помещён до её вызова.

6. Чем прототип функции отличается от ее заголовка?

Прототип – это объявление функции, не содержит тела функции. А заголовок – это часть описания функции, которая находится перед телом функции.

7. Каким образом включают в программу прототипы библиотечных функций?

При помощи директивы препроцессора `#include`. После директивы препроцессора указывается имя библиотеки или имя заголовочного файла библиотеки. Имя заключается в угловые скобки (для стандартных библиотек) или в двойные кавычки (для остальных библиотек).

8. Опишите передачу параметров в функцию по значению.

Когда аргумент используется в вызове по значению, то вызываемой функции передаётся копия значения аргумента. Изменения, происходящие с копией, не отражаются на значении исходной переменной в вызывающей функции.

9. Что такое макрос? Как создаются макросы?

Макрос – это операция, определяемая при помощи директивы препроцессора `#define`, перед компиляцией программы вместо идентификатора макроса в программу подставляется замещающий текст. Создаются следующим образом:
`#define ИМЯ_МАКРОСА [замещающий текст]`

10. Чем макросы отличаются от функций? В чем основная опасность использования макросов?

Макросы заменяют текст везде, а функции используются только во время вызова. Во время вызова макроса не проверяется тип, в то время как проверка типов выполняется во время вызова функции, также может возникнуть нежелательная замена.

11. Приведите определение рекурсивной функции. Чем рекурсия отличается от итерации?

Рекурсивная функция – это функция, которая вызывает саму себя или непосредственно, или через другую функцию. Итерация — организация обработки данных, при которой действия повторяются многократно, не приводя при этом к вызовам самих себя.

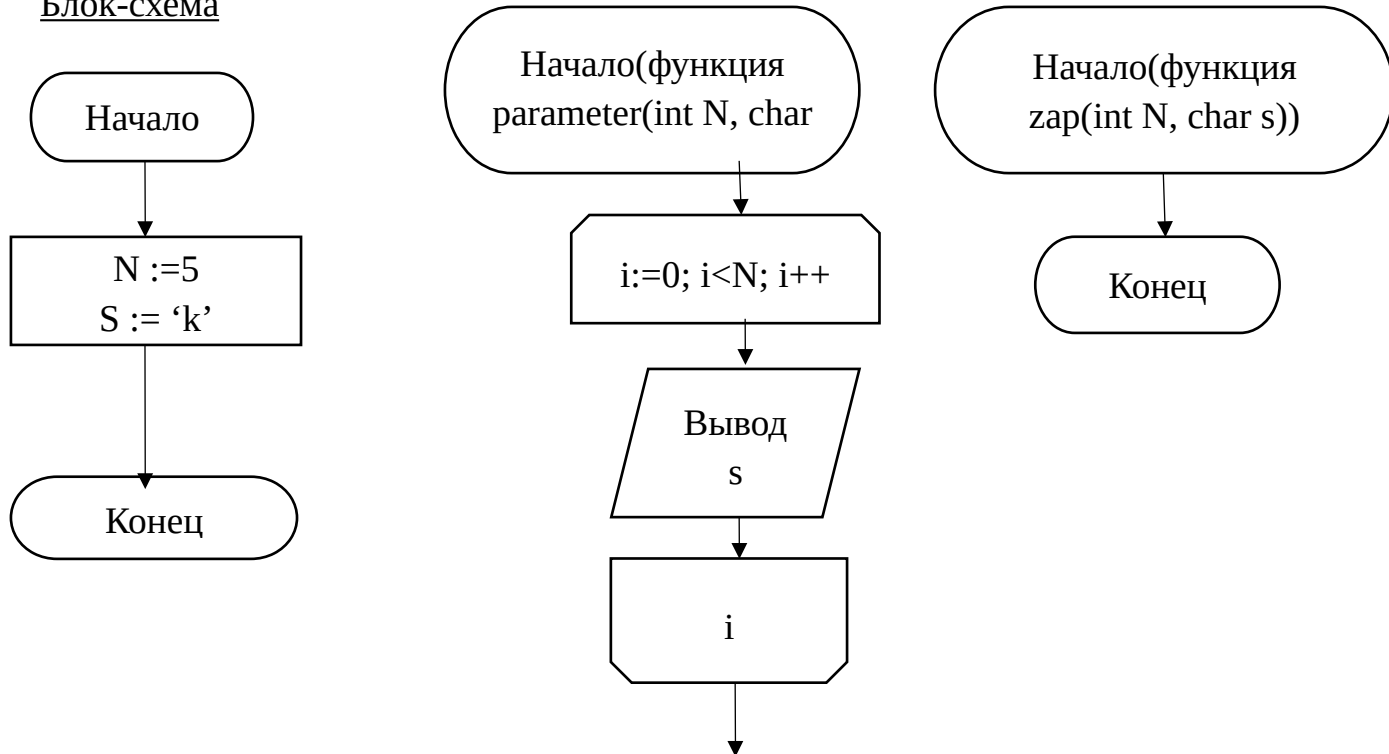
12. Какие элементы обязательно должен содержать рекурсивный алгоритм (рекурсивная функция)?

Рекурсивная функция должна содержать вызов самой себя.

Индивидуальное задание №1

Задание: Написать программу, содержащую две функции. Первая функция, выводящая на экран строку одинаковых символов, располагается до функции main(). Вторая, запрашивающая символ и количество символов в строке, располагается после функции main().

Блок-схема



Описание программы:

Задается функция, которая при помощи цикла for() выводит строку выбранных символов. Далее мы задаем сами параметры для данной функции в виде количества символов и сам символ.

Исходный код программы

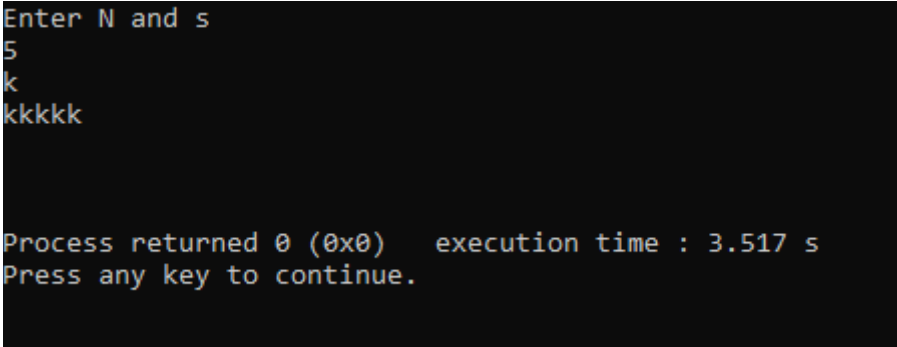
```
#include <stdio.h>
#include <string.h>

using namespace std;

int parameter(int N, char s)//инициализируем функцию
{
    int i;
    for (i = 0; i < N; i++)//цикл вывода символов
    {
        printf("%c", s);
    }
}
```

```
}  
return 0;  
}  
int zap();  
int main() {  
    int N;  
    char s;  
    zap();  
    scanf("%i %c", &N, &s);  
    parameter(N, s); //вводим параметры для функции  
    printf("\n\n\n");  
  
    return 0;  
}  
int zap(){  
    printf("Enter N and s");  
    return 0;  
}  
}
```

Пример выполнения программы



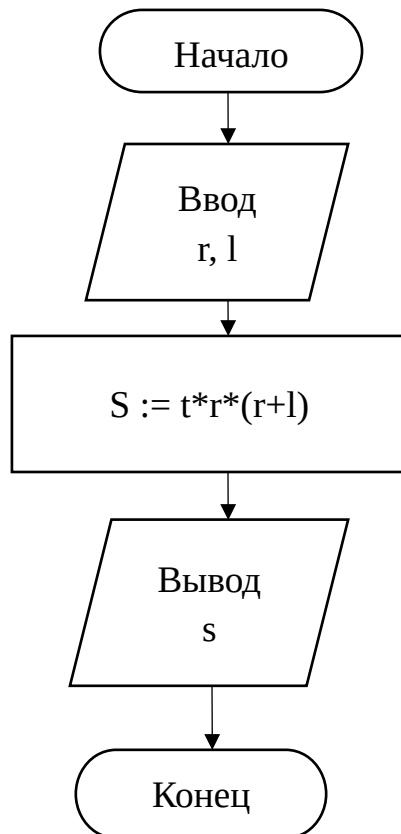
The screenshot shows a terminal window with a black background and white text. The first line is the prompt "Enter N and s". The user has entered "5" on the next line, "k" on the following line, and "kkkkk" on the line after that. Below these inputs, there is a blank line. At the bottom of the terminal, the text "Process returned 0 (0x0) execution time : 3.517 s" is displayed, followed by "Press any key to continue." on the next line.

```
Enter N and s  
5  
k  
kkkkk  
  
Process returned 0 (0x0) execution time : 3.517 s  
Press any key to continue.
```

Индивидуальное задание №2

Задание: Написать программу для вычисления площади полной поверхности конуса. Расчет площади организовать с помощью макроса препроцессора.

Блок-схема



Описание программы:

r – это радиус

l – это образующая

s - искомый объем

t – переменная равная по значению π

С помощью функции `define` задаем замещающийся текст для числа π и для объема конуса.

Вводим значения и находим объем

Исходный код программы

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

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

```
int l, r, t;
```

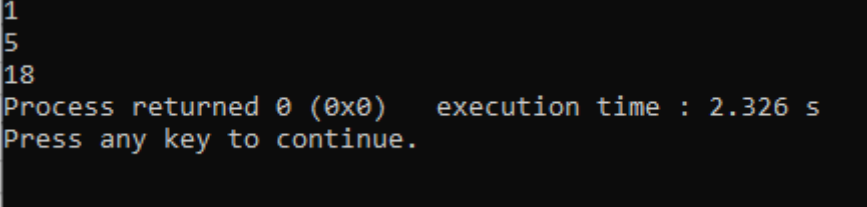
```
#define p 3.14 //замещаем число пи его значением
#define s t*r*(r+l) //замещаем символ объема его формулой

int main() {

    t = p;
    scanf("%i %i", &r, &l); //вводим радиус и образующую конуса
    printf("%i", s); //выводим объем

    return 0;
}
```

Пример выполнения программы



```
1
5
18
Process returned 0 (0x0)   execution time : 2.326 s
Press any key to continue.
```

Индивидуальное задание №3

Задание: Написать программу, которая вычисляет частичную сумму ряда

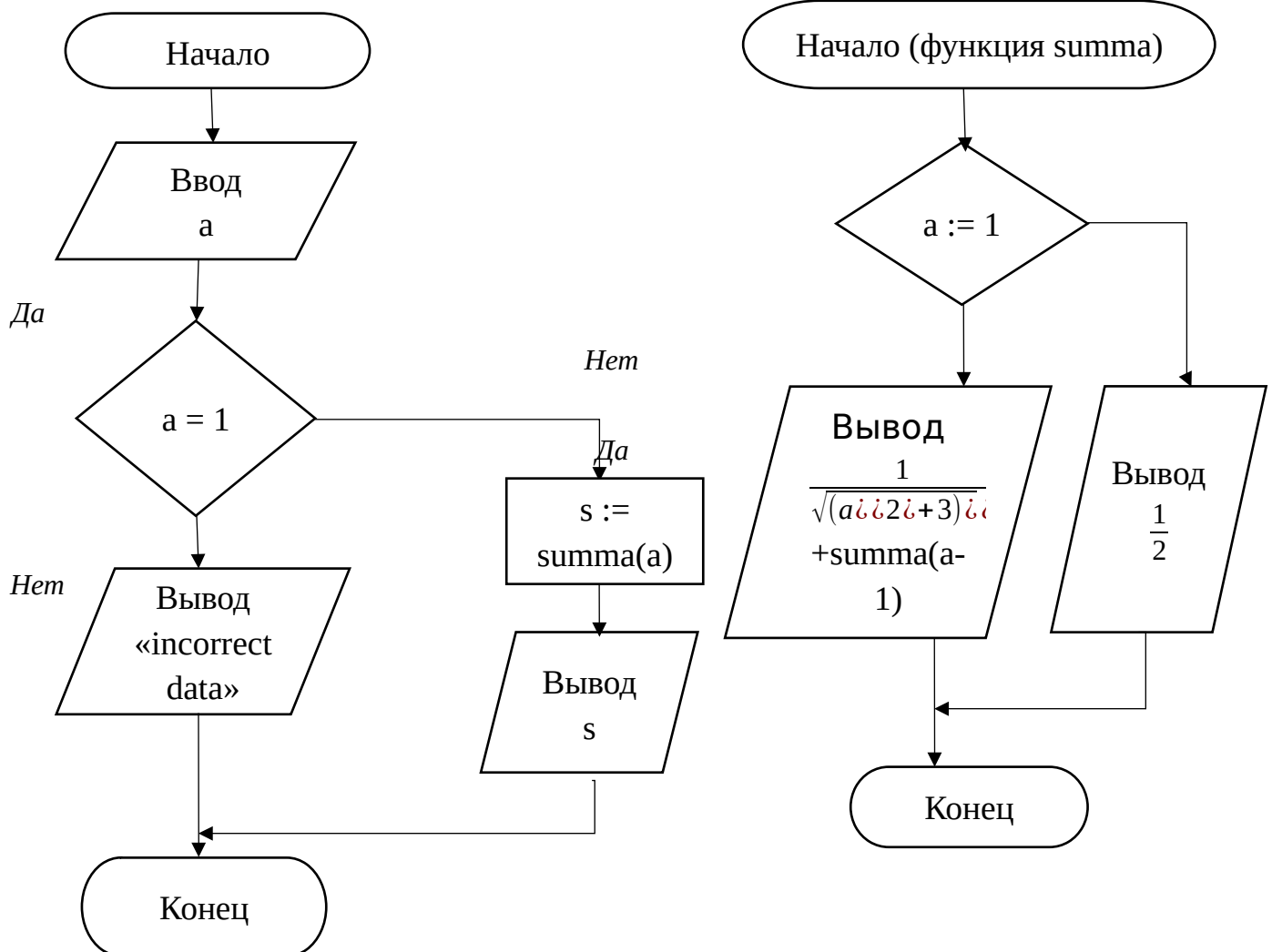
$$\sum_{k=1}^{\infty} \frac{1}{\sqrt{k^2 + 3}}$$

Расчет должен производиться двумя способами:

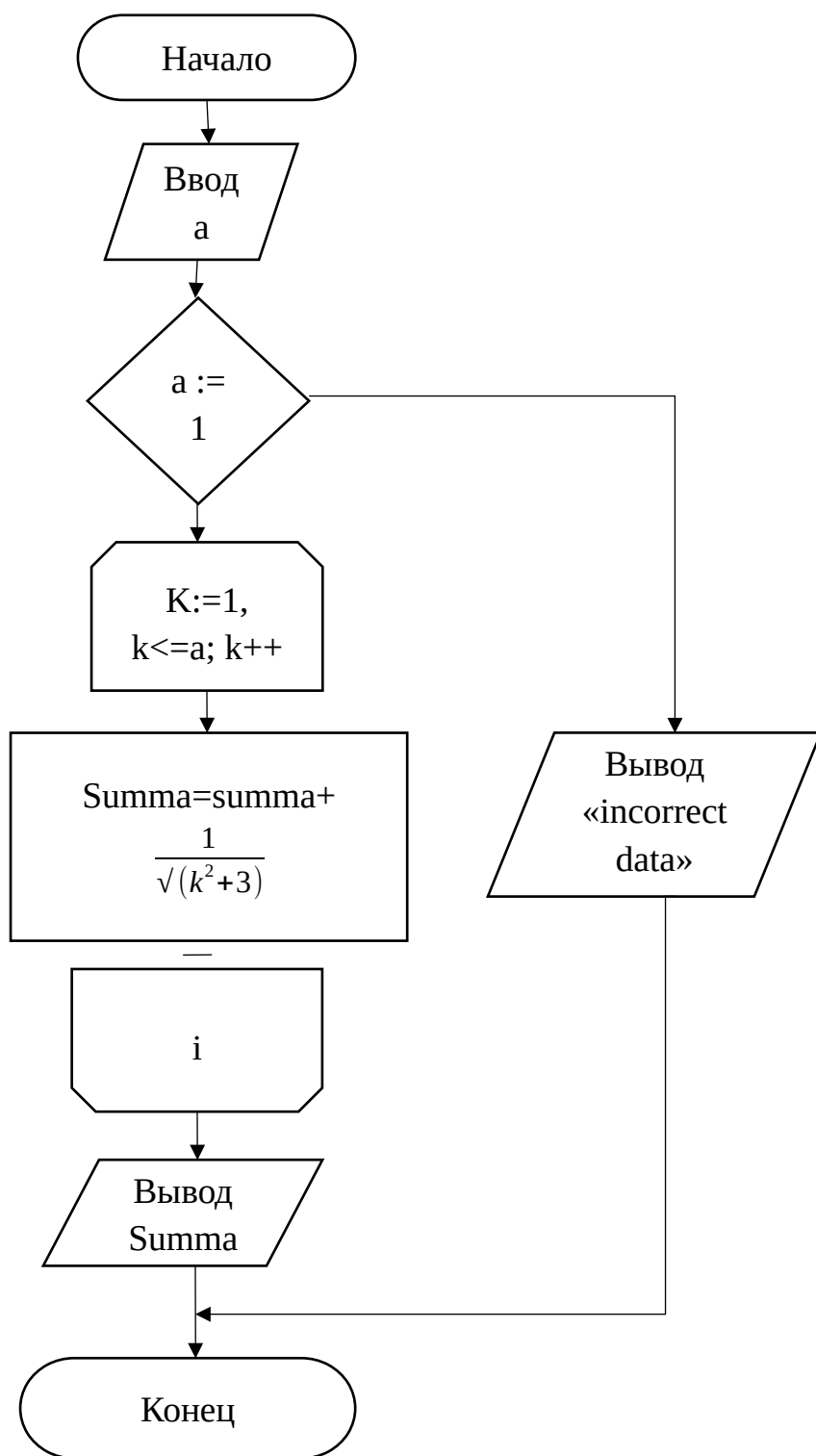
- при помощи итерационного цикла,
- при помощи рекуррентной формулы.

Блок-схема

Рекуррентная формула:



Итерация:



Описание программы:

1) Программа вычисляет частичную сумму ряда до вводимого пользователем числа. В основной программе происходит запрос переменной и ее проверка, затем идет обращение к дополнительной функции summa, которая

вычисляет сумму по требуемой формуле.

2) Программа вычисляет частичную сумму ряда до вводимого пользователем числа. Эта сумма вычисляется по требуемой формуле в цикле от 1 до определенного числа.

Исходный код программы

Рекурсия:

```
#ifndef FF_H_INCLUDED
#define FF_H_INCLUDED

float summa (int a) //инициализируем функцию
{
    if (a == 1) {return 1 / 2;} //выводим ответ при a=1
    else { return 1 / sqrt((double)(a*a + 3)) + summa(a-1);} //выводим частичную
самму данного выражения, при a не равном 1
}
#endif // FF_H_INCLUDED
```

```
#include <stdio.h>
#include <math.h>
#include "ff.h"
```

```
int main()
{
    int a;
    float s;
    printf ("entered value\n"); // просим пользователя ввести число
    if (scanf("%d", &a) == 1) // проверяем равно ли a = 1
    {
        s = summa(a);
        printf("%.2f", s); //выводим сумму
    }
    else {printf("incorrect data");} // если a не равно 1 выводим сообщение о
неверном значении
    return 0;
}
```

Итерация:

```

#include <stdio.h>
#include <math.h>
int main()
{
    int a, k=1;
    float summa = 0.0;
    printf ("entered value\n"); //просим пользователя ввести число
    if (scanf("%d", &a) == 1) //проверяем, введенное число равно 1
    {
        for (k = 1; k <= a; k++) // цикл от k до a включительно
        {
            summa += 1 / sqrt((double)(k*k + 3)); // суммируем все значения
        }
        printf("%.2f", summa); //выводим получившееся частное суммы от данного
выражения
    }
    else { printf("incorrect data"); } //иначе выводим сообщение о неверном
значении
    return 0;
}

```

Пример выполнения программы

```

entered value
10
4.02
Process returned 0 (0x0)   execution time : 2.195 s
Press any key to continue.

```

Вывод

В процессе выполнения лабораторной работы были расширены знания в таких понятиях как макрос-объект, макрос-функция. Ознакомлен со стандартными библиотеками языка Си, обучен созданию собственной библиотеки, принципам разбиения программ на блоки и правилам оформления функций на языке Си.

Список использованной литературы

1. Дейтел Х. М. Как программировать на С.
2. Информация с сайта <http://bikmeyev-at.ugatu.su/students/CPP/index.html> .