

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ им. В. Г. Шухова»
(БГТУ им. В. Г. Шухова)**

Кафедра программного обеспечения вычислительной
техники и автоматизированных систем

Лабораторная работа №19.11
по дисциплине: «Рекурсивные функции»

Выполнил/а: ст. группы ВТ-231
Кисиль Николай Владимирович

Проверили:
Черников Сергей Викторович
Новожен Никита Викторович

Белгород, 2023 г.

Цель работы: получение навыков написания рекурсивных функций

Содержание работы

Задача 1: Определить количество цифр в тексте, вводимом с клавиатуры. Текст заканчивается символом перехода на новую строку \n.....	4
Задача 2: Вывести данное натуральное число в восьмеричной системе счисления	5
Задача 3: Дан знаменатель и первый член геометрической прогрессии. Вычислить n -й член прогрессии.....	6
Задача 4: Дана упорядоченная по убыванию последовательность целых чисел. Определить, есть ли среди членов данной последовательности число x , и если есть, найти номер этого члена.....	7
Задача 5: Дан массив a размера n ($n \geq 2$). Необходимо проверить, является ли он упорядоченным по неубыванию.	8
Задача 6: Вывести данное натуральное число в восьмеричной системе счисления	9
Задача 7: Даны натуральные числа a и b . Определить, могут ли эти числа быть соседними членами последовательности Фибоначчи.....	10
Задача 8: Вывести в обратном порядке символы данного текста, вводимого с клавиатуры, которые не являются цифрами. Текст заканчивается символом перехода на новую строку \n.	11
Задача 9: Дан n -й член арифметической прогрессии, ее разность и значение n . Вычислить первый член прогрессии.	12
Задача 10: С клавиатуры вводятся положительные вещественные числа a_1, a_2, \dots, a_n . Признаком конца ввода – отрицательное число.....	13
Задача 11: Реализовать функции any , которая возвращает значение 'истина', если хотя бы один элемент удовлетворяет функции-предикату f , в	

противном случае – ложь. Реализовать функцию all, которая возвращает значение 'истина', если все элементы удовлетворяет функции-предикату f, в противном случае – ложь..... 14

Задача 12: Реализовать алгоритм бинарного поиска 15

Задача 13: Реализовать сортировку выбором 16

Задача 1: Определить количество цифр в тексте, вводимом с клавиатуры. Текст заканчивается символом перехода на новую строку \n.

Код:

```
int getCountDigitsInText(char s, int count) {
    s = getchar();
    if (s == '\n') {
        return count;
    } else {
        count += isdigit(s);
        return getCountDigitsInText(s, count);
    }
}

int getCountDigits(char s) {
    return getCountDigitsInText(s, 0);
}
```

Задача 2: Вывести данное натуральное число в восьмеричной системе счисления

Код:

```
void printOct(int x) {  
    if (x) {  
        printOct(x / 8);  
        printf("%d", x % 8);  
    }  
}
```

Задача 3: Дан знаменатель и первый член геометрической прогрессии. Вычислить n -й член прогрессии

Код:

```
int getMemberGeometricProg(int a1, int q, int n) {  
    if (n > 2) {  
        return getMemberGeometricProg(a1 * q, q, --n);  
    } else {  
        return a1 * q;  
    }  
}
```

Задача 4: Дана упорядоченная по убыванию последовательность целых чисел. Определить, есть ли среди членов данной последовательности число x , и если есть, найти номер этого члена.

Код:

```
int getIndex(int *a, int n, int x) {  
    if (a[n - 1] != x && n) {  
        return getIndex(a, --n, x);  
    } else {  
        return n - 1;  
    }  
}
```

Задача 5: Дан массив a размера n ($n \geq 2$). Необходимо проверить, является ли он упорядоченным по неубыванию.

Код:

```
int sortedUnDecreasing(int *a, int n) {  
    if (a[n - 2] < a[n - 1] && n >= 3) {  
        return sortedUnDecreasing(a, --n);  
    } else {  
        return a[n - 2] < a[n - 1];  
    }  
}
```


Задача 6: Найти номер первого вхождения минимального значения в последовательность длины n (линейный поиск)

Код:

```
int firstMin(int *a, int n, int cur_index, int min_index) {
    if (cur_index == n) {
        return min_index;
    }

    if (a[cur_index] < a[min_index]) {
        min_index = cur_index;
    }
    return firstMin(a, n, cur_index + 1, min_index);
}

int getFirstMin(int *a, int n) {
    return firstMin(a, n, 1, 0);
}
```

Задача 7: Даны натуральные числа a и b . Определить, могут ли эти числа быть соседними членами последовательности Фибоначчи.

Код:

```
int isFibonacci(int sum, int a, int b) {  
    if (b == sum) {  
        return 1;  
    } else if (b > sum) {  
        return 0;  
    } else {  
        return isFibonacci(sum, b, a + b);  
    }  
}  
  
int isFibonacciNeighbor(int a, int b) {  
    if(a < 1 || b < 1)  
        return 0;  
    int sum = a + b;  
    return isFibonacci(sum, 1, 1);  
}
```

Задача 8: Вывести в обратном порядке символы данного текста, вводимого с клавиатуры, которые не являются цифрами. Текст заканчивается символом перехода на новую строку \n.

Код:

```
void printRevers(char s) {  
    s = getchar();  
    if (s != '\n') {  
        printRevers(s);  
    }  
    if (isdigit(s) == 0) {  
        printf("%c", s);  
    }  
}
```

Задача 9: Дан n -й член арифметической прогрессии, ее разность и значение n . Вычислить первый член прогрессии.

Код:

```
int getFirstArifMean(int n, int d) {  
    if (n > d) {  
        return getFirstArifMean(n - d, d);  
    } else {  
        return n;  
    }  
}
```

Задача 10: С клавиатуры вводятся положительные вещественные числа a_1, a_2, \dots, a_n . Признаком конца ввода – отрицательное число.

Код:

```
void task10(float a1, float a2) {  
    float result = (a1 + a2) / 2;  
    printf("%f ", result);  
  
    if (a2 >= 0) {  
        float new_a2;  
        scanf("%f", &new_a2);  
        task10(a2, new_a2);  
    }  
}
```

Задача 11: Реализовать функции any, которая возвращает значение 'истина', если хотя бы один элемент удовлетворяет функции-предикату f, в противном случае – ложь. Реализовать функцию all, которая возвращает значение 'истина', если все элементы удовлетворяет функции-предикату f, в противном случае – ложь

Код:

```
int isPositive(int x) {
    return x > 0;
}

int anyPredicate(int *a, int n, int (*predicate)(int)) {
    if (n == 0) {
        return 0;
    } else {
        return predicate(a[0]) || anyPredicate(a + 1, n - 1, predicate);
    }
}

int allPredicate(int *a, int n, int (*predicate)(int)) {
    if (n == 0) {
        return 1;
    } else {
        return predicate(a[0]) && allPredicate(a + 1, n - 1, predicate);
    }
}
```

Задача 12: Реализовать алгоритм бинарного поиска

Код:

```
int binarySearch(int *a, int l, int r, int x) {
    if (l <= r) {
        int m = (l + r) / 2;

        if (a[m] == x) {
            return m;
        }

        if (a[m] > x) {
            return binarySearch(a, l, m - 1, x);
        }

        return binarySearch(a, m + 1, r, x);
    }

    return -1;
}
```

Задача 13: Реализовать сортировку выбором

Код:

```
void selectionSort(int *a, int n) {  
    if (n) {  
        int min_index = getFirstMin(a, n);  
        swap(&a[0], &a[min_index]);  
  
        selectionSort(a + 1, --n);  
    }  
}
```

Вывод: реализовали решение задач с помощью рекурсивных функций, получили навыки написания рекурсивных функций