|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| *Федеральное государственное бюджетное образовательное учреждение высшего образования* ***«МИРЭА – Российский технологический университет»***  **РТУ МИРЭА** |

**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**Дисциплина «Программирование на языке Джава»**

**ОТЧЕТ**

**ПО ПРАКТИЧЕСКОМУ ЗАНЯТИЮ №10**

Выполнил студент группы ИВБО-07-19 Гудаев И.И.

Принял Степанин П.С.

Практические работы выполнены «\_\_\_»\_\_\_\_\_\_\_2020г.

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_» «\_\_\_»\_\_\_\_\_\_\_2020г.

Отметка о выполнении

**Москва – 2020 г.**

## **Задание**

1. Треугольная последовательность. Дана монотонная последовательность, в которой каждое натуральное число k встречается ровно k раз: 1, 2, 2, 3, 3, 3, 4, 4, 4, 4,… По данному натуральному n выведите первые n членов этой последовательности. Попробуйте обойтись только одним циклом for.  
2. От 1 до n. Дано натуральное число n. Выведите все числа от 1 до n.  
3. От A до B. Даны два целых числа A и В (каждое в отдельной строке). Выведите все числа от A до B включительно, в порядке возрастания, если A < B, или в порядке убывания в противном случае.  
4. Заданная сумма цифр. Даны натуральные числа k и s. Определите, сколько существует k-значных натуральных чисел, сумма цифр которых равна d. Запись натурального числа не может начинаться с цифры 0. В этой задаче можно использовать цикл для перебора всех цифр, стоящих на какой-либо позиции.  
5. Сумма цифр числа. Дано натуральное число N. Вычислите сумму его цифр. При решении этой задачи нельзя использовать строки, списки, массивы (ну и циклы, разумеется).  
6. Проверка числа на простоту. Дано натуральное число n>1. Проверьте, является ли оно простым. Программа должна вывести слово YES, если число простое и NO, если число составное. Алгоритм должен иметь сложность O(logn). Указание: понятно, что задача сама по себе является нерекурсивной, т.к. проверка числа n на простоту никак не сводится к проверке на простоту меньших чисел. Поэтому нужно сделать еще один параметр рекурсии: делитель числа, и именно по этому параметру и делать рекурсию.  
7. Разложение на множители. Дано натуральное число n>1. Выведите все простые множители этого числа в порядке неубывания с учетом кратности. Алгоритм должен иметь сложность O(logn)  
8. Палиндром. Дано слово, состоящее только из строчных латинских букв. Проверьте, является ли это слово палиндромом. Выведите YES или NO. При решении этой задачи нельзя пользоваться циклами, в решениях на питоне нельзя использовать срезы с шагом, отличным от 1.  
9. Без двух нулей. Даны числа a и b. Определите, сколько существует последовательностей из a нулей и b единиц, в которых никакие два нуля не стоят рядом.  
10. Разворот числа. Дано число n, десятичная запись которого не содержит нулей. Получите число, записанное теми же цифрами, но в противоположном порядке. При решении этой задачи нельзя использовать циклы, строки, списки, массивы, разрешается только рекурсия и целочисленная арифметика. Функция должна возвращать целое число, являющееся результатом работы программы, выводить число по одной цифре нельзя.  
11. Количество единиц. Дана последовательность натуральных чисел (одно число в строке), завершающаяся двумя числами 0 подряд. Определите, сколько раз в этой последовательности встречается число 1. Числа, идущие после двух нулей, необходимо игнорировать. В этой задаче нельзя использовать глобальные переменные и параметры, передаваемые в функцию. Функция получает данные, считывая их с клавиатуры, а не получая их в виде параметров.  
12. Вывести нечетные числа последовательности. Дана последовательность натуральных чисел (одно число в строке), завершающаяся числом 0. Выведите все нечетные числа из этой последовательности, сохраняя их порядок. В этой задаче нельзя использовать глобальные переменные и передавать какие-либо параметры в рекурсивную функцию. Функция получает данные, считывая их с клавиатуры. Функция не возвращает значение, а сразу же выводит результат на экран. Основная программа должна состоять только из вызова этой функции.  
13. Вывести члены последовательности с нечетными номерами. Дана последовательность натуральных чисел (одно число в строке), завершающаяся числом 0. Выведите первое, третье, пятое и т.д. из введенных чисел. Завершающий ноль выводить не надо. В этой задаче нельзя использовать глобальные переменные и передавать какие-либо параметры в рекурсивную функцию. Функция получает данные, считывая их с клавиатуры. Функция не возвращает значение, а сразу же выводит результат на экран. Основная программа должна состоять только из вызова этой функции.  
14. Цифры числа слева направо. Дано натуральное число N. Выведите все его цифры по одной, в обычном порядке, разделяя их пробелами или новыми строками. При решении этой задачи нельзя использовать строки, списки, массивы (ну и циклы, разумеется). Разрешена только рекурсия и целочисленная арифметика  
15. Цифры числа справа налево. Дано натуральное число N. Выведите все его цифры по одной, в обратном порядке, разделяя их пробелами или новыми строками. При решении этой задачи нельзя использовать строки, списки, массивы (ну и циклы, разумеется). Разрешена только рекурсия и целочисленная арифметика.  
16. Количество элементов, равных максимуму. Дана последовательность натуральных чисел (одно число в строке), завершающаяся числом 0. Определите, какое количество элементов этой последовательности, равны ее наибольшему элементу. В этой задаче нельзя использовать глобальные переменные. Функция получает данные, считывая их с клавиатуры, а не получая их в виде параметра. В программе на языке Python функция возвращает результат в виде кортежа из нескольких чисел, и функция вообще не получает никаких параметров. В программе на языке Java результат записывается в переменные, которые передаются в функцию по ссылке. Других параметров, кроме как используемых для возврата значения, функция не получает. Гарантируется, что последовательность содержит хотя бы одно число (кроме нуля)  
17. Максимум последовательности. Дана последовательность натуральных чисел (одно число в строке), завершающаяся числом 0. Определите наибольшее значение числа в этой последовательности. В этой задаче нельзя использовать глобальные переменные и передавать какие-либо параметры в рекурсивную функцию. Функция получает данные, считывая их с клавиатуры. Функция возвращает единственное значение: максимум считанной последовательности. Гарантируется, что последовательность содержит хотя-бы одно число (кроме нуля).

## **Ход Работы**

В ходе выполнения работы были получены следующие исходные коды:

**package** com.company;

**import** java.util.Random;

**import** java.util.Scanner;

**public** **class** Main {

**static** **Scanner** sc = **new** **Scanner**(**System**.in);

**public** **static** **String** triangSeq(int n) //�-адание 1. Треугольная последовательность.

{

int s = 0;

int j = 0;

**if** (n == 1){

**System**.out.print("1");

}

**else** {

**for** (int i = 1; s < n; i++) {

s += i;

j = i;

}

**System**.out.print(triangSeq(--n) + " " + j);

}

**return** "";

}

**public** **static** void oneToN()

{

int n;

**System**.out.println("�'ведите число n: ");

n = sc.nextInt();

**for** (int i = 1;i <= n; i++) **System**.out.print(i + " ");

}

**public** **static** void AtoB(){

int A, B;

**System**.out.println("�'ведите число A.");

A = sc.nextInt();

**System**.out.println("�'ведите число B.");

B = sc.nextInt();

**if**(A<B){

**for**(int i = A; i <= B; i++) **System**.out.print(i + " ");

}

**else** **if**(A>B){

**for**(int i = A; i >= B; i--) **System**.out.print(i + " ");

}

**else** **System**.out.println("A равно B.");

}

**public** **static** void sumOfDigits(){

int k, s, sum = 0, tmp, count = 0;

**System**.out.println("�'ведите число k.");

k = sc.nextInt();

**System**.out.println("�'ведите число s.");

s = sc.nextInt();

**for**(int i = (int)**Math**.pow(10, k - 1); i<(int)**Math**.pow(10, k); i++){

tmp = i;

**while**(tmp>0){

sum += (tmp%10);

tmp /= 10;

}

**if**(sum == s) count++;

sum = 0;

Листинг 1.1. Задание 1 (класс Main)

}

**System**.out.println("Cуществует " + count + " k-значных натуральных чисел, сумма цифр которых равна s");

}

**public** **static** int sumOfNumbersDigits(int num){

**if**(num <= 0) **return** num;

**return** num % 10 + sumOfNumbersDigits(num / 10);

}

**public** **static** void isSimple(long num, long cons) {

**if** (num > 1) {

**if** (num % cons != 0) {

cons++;

isSimple(num, cons);

}

**else** **if** (num % cons == 0) {

**if** (cons == num) **System**.out.println("YES");

**else** **System**.out.println("NO");

}

}

}

**public** **static** void factorization(int num, int cons){

**if** (num % cons == 0 & cons != num) {

**System**.out.print(cons + " ");

factorization(num / cons, cons);

}

**else**{

**if**(cons == num) **System**.out.println(cons);

**else**{

cons++;

factorization(num, cons);

}

}

}

**public** **static** void palindrome(){

**String** str = **new** **String**();

sc.nextLine();

str = (sc.nextLine());

**if**(str.equals(**new** **StringBuffer**(str).reverse().toString())) **System**.out.println("YES");

**else** **System**.out.println("NO");

}

**public** **static** int withoutTwoZeros(int a, int b){

**if**(a > b + 1) **return** 0;

**if**(a == 0 || b ==0) **return** 1;

**return** withoutTwoZeros(a, b-1) + withoutTwoZeros(a-1, b-1);

}

**public** **static** int numberReversal(int num, int b){

**if**(num == 0) **return** b;

**else**{

b = b \* 10 + num % 10;

num /= 10;

**return** numberReversal(num, b);

}

}

**public** **static** void amOfUnits(){

int fir = sc.nextInt();

int sec = sc.nextInt();

int count = 0;

**while**(fir != 0 || sec != 0){

**if**(fir == 1) count++;

fir = sec;

sec = sc.nextInt();

}

**System**.out.println(count);

}

Листинг 1.2. Задание 1 (класс Main)

**public** **static** void oddSeqNums(){

int[] arr = **new** int[100];

int k = 0;

int tmp = sc.nextInt();

**while**(tmp != 0){

**if**(tmp % 2 != 0) {

arr[k] = tmp;

k++;}

tmp = sc.nextInt();

}

**for** (int i = 0;i<k;i++) **System**.out.print(arr[i] + " ");

}

**public** **static** void oddNumberedSequenceMembers(){

int[] arr = **new** int[100];

int k = 0;

int tmp = sc.nextInt();

**while**(tmp != 0){

arr[k] = tmp;

k++;

tmp = sc.nextInt();

}

**for** (int i = 0;i < k;i += 2) **System**.out.print(arr[i] + " ");

}

**public** **static** **String** digitsNumbersFromLeftToRight(int num){

**if** (num < 10){

**return** **Integer**.toString(num);

}

**else**{

**return** digitsNumbersFromLeftToRight(num / 10) + " " + num % 10;

}

}

**public** **static** int digitsNumbersFromRightToLeft(int num){

**if** (num <= 0) {

**return** num;

}

**else** {

**System**.out.print(num % 10 + " ");

**return** digitsNumbersFromRightToLeft(num / 10);

}

}

**public** **static** void numberOfElementsEqualToMaximum(int max, int count) {

int n = sc.nextInt();

**if** (n > 0){

**if** (n > max){

numberOfElementsEqualToMaximum(n, 1);

}

**else** **if** (n == max){

numberOfElementsEqualToMaximum(max, ++count);

}

**else**{

numberOfElementsEqualToMaximum(max, count);

}

}

**else**{

**System**.out.println(count);

}

}

**public** **static** int maximumConsistency(){

int num = sc.nextInt();

**if** (num == 0) {

**return** 0;

}

**else** {

**return** **Math**.max(num, maximumConsistency());

}

}

Листинг 1.3. Задание 1 (класс Main)

**public** **static** void main(**String**[] args) {

**System**.out.println("задание 1. Треугольная последовательность.");

triangSeq(sc.nextInt());

**System**.out.println("\nзадание 2. От 1 до n.");

oneToN();

**System**.out.println("\nзадание 2. От A до B.");

AtoB();

**System**.out.println("\nзадание 4. заданная сумма цифр.");

sumOfDigits();

**System**.out.println("задание 5. Сумма цифр числа.");

**System**.out.println(sumOfNumbersDigits(sc.nextInt()));

**System**.out.println("задание 6. Проверка числа на простоту.");

isSimple(sc.nextInt(), 2);

**System**.out.println("задание 7. Разложение на множители.");

factorization(sc.nextInt(),2);

**System**.out.println("задание 8. Палиндром.");

palindrome();

**System**.out.println("задание 9. без двух нулей.");

**System**.out.println(withoutTwoZeros(sc.nextInt(), sc.nextInt()));

**System**.out.println("задание 10. Разворот числа.");

**System**.out.println(numberReversal(sc.nextInt(), 0));

**System**.out.println("задание 11. Количество единиц.");

amOfUnits();

**System**.out.println("задание 12. Нечетные числа последовательности.");

oddSeqNums();

**System**.out.println("\nзадание 13. Члены последовательности с нечетными номерами.");

oddNumberedSequenceMembers();

**System**.out.println("\nзадание 14. Цифры числа слева направо.");

**System**.out.println(digitsNumbersFromLeftToRight(sc.nextInt()));

**System**.out.println("задание 15. Цифры числа справа налево.");

digitsNumbersFromRightToLeft(sc.nextInt());

**System**.out.println("\nзадание 16. Количество элементов, равных максимуму.");

numberOfElementsEqualToMaximum(0,0);

**System**.out.println("\n задание 17. Максимум последовательности.");

**System**.out.println(maximumConsistency());

}

}

Листинг 1.4. Задание 1 (класс Dish)

## **Вывод**

В этой работе мы использовали рекурсивные функции для решения множества задач, разобрались в понятии «сложность алгоритма O(n)» и написали функции, имеющие определенную сложность