

Недобежкин Павел Владимирович

ПММ, 4 курс, 61 группа

Отчет по лабораторной работе №1

Задание 1.

Входные данные: Два натуральных числа $n \leq 40$ и $k \leq 5$. Число k - число рождающихся пар крольчат у пары кроликов, n - число месяцев, в течении которых размножаются кролики.

Выходные данные: Общее число пар кроликов которое образуется после n месяцев если изначально присутствует только одна пара и в каждом поколении каждая пара репродуктивного возраста производит потомство, состоящее из k пар кроликов (вместо одной пары из последовательности Фибоначчи).

Пример входных данных: $n = 5, k = 3$

Пример выходных данных: 19

Код программы:

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n, k;

        System.out.print("Введите количество месяцев: ");
        n = sc.nextInt();
        System.out.print("Введите количество рождающихся пар: ");
        k = sc.nextInt();

        System.out.println("Количество пар: " + fibfunc(n,k));
    }

    public static int fibfunc(int n, int k){
        int n_prev = 0;
        int n_cur = 1;
        int n_fut = 0;

        for(int i = 0; i < n-1; i++){
            n_fut = n_cur + n_prev*k;
            n_prev = n_cur;
            n_cur = n_fut;
        }

        return n_cur;
    }
}
```

Результат: 19

Задание 2.

Входные данные: s - последовательность ДНК.

Выходные данные: четыре целых числа, разделённые пробелами, представляющие собой количество символов 'A', 'C', 'G' и 'T', содержащихся в s.

Пример входных данных:

AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAGAGTGTCTGATAGCAGC

Пример выходных данных: 20 12 17 21

Код программы:

```
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String n;

        System.out.print("Введите последовательность: ");
        n = sc.nextLine();

        int[] res = count_func(n);
        System.out.println("Результат: ");
        for(int i = 0; i < 4; i++){
            System.out.print(res[i] + " ");
        }
    }

    public static int[] count_func(String n){
        int[] res = new int[4];

        res[0] = n.length() - n.replace("A", "").length();
        res[1] = n.length() - n.replace("C", "").length();
        res[2] = n.length() - n.replace("G", "").length();
        res[3] = n.length() - n.replace("T", "").length();

        return res;
    }
}
```

Результат: 20 12 17 21

Задание 3. Даны две строки s и t одинаковой длины. Найти расстояние Хэмминга $dH(s,t)$ между ними.

Входные данные: две последовательности ДНК s и t одинаковой длины

Выходные данные: расстояние Хэмминга $dH(s,t)$

Пример входных данных:

GAGCCTACTAACGGGAT

CATCGTAATGACGGCCT

Пример выходных данных: 7

Код программы:

```
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String s, t;

        System.out.print("Введите первую строку: ");
        s = sc.nextLine();

        System.out.print("Введите вторую строку: ");
        t = sc.nextLine();

        int res = hamming_distance(s, t);
        System.out.println("Результат: " + res);
    }

    public static int hamming_distance(String s, String t){
        int res = 0;
        if(s.length()!=t.length()){
            System.out.println("Строки разной длины");
            res = -1;
        }
        else{
            for(int i = 0; i < s.length(); i++){
                if(s.charAt(i) != t.charAt(i)){
                    res++;
                }
            }
        }
        return res;
    }
}
```

Результат: 7

Задание 4. Дана последовательность ДНК t. Заменяя в t тимин 'Т' на урацил 'У'.

Входные данные: последовательность ДНК.

Выходные данные: последовательность, полученная из t заменой 'Т' на 'У'.

Пример входных данных: GATGGAAC TTGACTACGTAAAT

Пример выходных данных: GAUGGAACUUGACUACGUAAAUU

Код программы:

```
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String s;
```

```

        System.out.print("Введите строку: ");
        s = sc.nextLine();

        String res = replacement(s);
        System.out.println("Результат: "+ res);
    }

    public static String replacement(String s){
        String res = s.replace("T", "U");
        return res;
    }
}

```

Результат: GAUGGAACUUGACUACGUAAAUU

Задание 5.

Входные данные: последовательность мРНК t

Выходные данные: соответствующая t белковая последовательность s

Пример входных данных:

AUGGCCAUGGCGCCCAGAACUGAGAUCAAUAGUACCCGUUUUAACGGGUGA

Пример выходных данных: MAMAPRTEINSTRING

Код программы:

```

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String s;

        System.out.print("Введите строку: ");
        s = sc.nextLine();

        String res = convert(s);
        System.out.println("Результат: "+ res);
    }

    public static final String[][][] CODON_TABLE = new String[4][4][4];

    static {
        CODON_TABLE[0][0][0] = "F"; // UUU
        CODON_TABLE[0][0][1] = "F"; // UUC
        CODON_TABLE[0][0][2] = "L"; // UUA
        CODON_TABLE[0][0][3] = "L"; // UUG
    }
}

```

```
CODON_TABLE[0][1][0] = "S"; // UCU
CODON_TABLE[0][1][1] = "S"; // UCC
CODON_TABLE[0][1][2] = "S"; // UCA
CODON_TABLE[0][1][3] = "S"; // UCG

CODON_TABLE[0][2][0] = "Y"; // UAU
CODON_TABLE[0][2][1] = "Y"; // UAC
CODON_TABLE[0][2][2] = "Stop"; // UAA
CODON_TABLE[0][2][3] = "Stop"; // UAG

CODON_TABLE[0][3][0] = "C"; // UGU
CODON_TABLE[0][3][1] = "C"; // UGC
CODON_TABLE[0][3][2] = "Stop"; // UGA
CODON_TABLE[0][3][3] = "W"; // UGG

CODON_TABLE[1][0][0] = "L"; // CUU
CODON_TABLE[1][0][1] = "L"; // CUC
CODON_TABLE[1][0][2] = "L"; // CUA
CODON_TABLE[1][0][3] = "L"; // CUG

CODON_TABLE[1][1][0] = "P"; // CCU
CODON_TABLE[1][1][1] = "P"; // CCC
CODON_TABLE[1][1][2] = "P"; // CCA
CODON_TABLE[1][1][3] = "P"; // CCG

CODON_TABLE[1][2][0] = "H"; // CAU
CODON_TABLE[1][2][1] = "H"; // CAC
CODON_TABLE[1][2][2] = "Q"; // CAA
CODON_TABLE[1][2][3] = "Q"; // CAG

CODON_TABLE[1][3][0] = "R"; // CGU
CODON_TABLE[1][3][1] = "R"; // CGC
CODON_TABLE[1][3][2] = "R"; // CGA
CODON_TABLE[1][3][3] = "R"; // CGG

CODON_TABLE[2][0][0] = "I"; // AUU
CODON_TABLE[2][0][1] = "I"; // AUC
CODON_TABLE[2][0][2] = "I"; // AUA
CODON_TABLE[2][0][3] = "M"; // AUG

CODON_TABLE[2][1][0] = "T"; // ACU
CODON_TABLE[2][1][1] = "T"; // ACC
CODON_TABLE[2][1][2] = "T"; // ACA
CODON_TABLE[2][1][3] = "T"; // ACG

CODON_TABLE[2][2][0] = "N"; // AAU
CODON_TABLE[2][2][1] = "N"; // AAC
CODON_TABLE[2][2][2] = "K"; // AAA
CODON_TABLE[2][2][3] = "K"; // AAG

CODON_TABLE[2][3][0] = "S"; // AGU
```

```

        CODON_TABLE[2][3][1] = "S"; // AGC
        CODON_TABLE[2][3][2] = "R"; // AGA
        CODON_TABLE[2][3][3] = "R"; // AGG

        CODON_TABLE[3][0][0] = "V"; // GUU
        CODON_TABLE[3][0][1] = "V"; // GUC
        CODON_TABLE[3][0][2] = "V"; // GUA
        CODON_TABLE[3][0][3] = "V"; // GUG

        CODON_TABLE[3][1][0] = "A"; // GCU
        CODON_TABLE[3][1][1] = "A"; // GCC
        CODON_TABLE[3][1][2] = "A"; // GCA
        CODON_TABLE[3][1][3] = "A"; // GCG

        CODON_TABLE[3][2][0] = "D"; // GAU
        CODON_TABLE[3][2][1] = "D"; // GAC
        CODON_TABLE[3][2][2] = "E"; // GAA
        CODON_TABLE[3][2][3] = "E"; // GAG

        CODON_TABLE[3][3][0] = "G"; // GGU
        CODON_TABLE[3][3][1] = "G"; // GGC
        CODON_TABLE[3][3][2] = "G"; // GGA
        CODON_TABLE[3][3][3] = "G"; // GGG
    }

    public static String convert(String s){
        String res = "";
        String cur_prot;
        String codon_to_ind = "UCAG";
        for(int i = 0; i < s.length(); i += 3){
            cur_prot =
CODON_TABLE[codon_to_ind.indexOf(s.charAt(i))][codon_to_ind.indexOf(s.charAt(i+1)
)][codon_to_ind.indexOf(s.charAt(i+2))];
            if(cur_prot == "Stop"){
                break;
            }
            res += cur_prot;
        }

        return res;
    }
}

```

Результат: MAMAPRTEINSTRING

Вывод:

В ходе лабораторной работы мы научились решать простейшие задачи, повторили последовательность Фибоначчи и научились работать со строками.