Содержание

[1 Постановка задачи 3](#_Toc116916030)

[1.1 Исходное условие задачи 3](#_Toc116916031)

[1.2 Модифицированное условие задачи 3](#_Toc116916032)

[1.3 Исходные данные 3](#_Toc116916033)

[2 Структура данных 4](#_Toc116916034)

[3 Детали и методика решения задачи 5](#_Toc116916035)

[3.1 Детали задачи 5](#_Toc116916036)

[3.2 Используемые формулы 5](#_Toc116916037)

[3.3 Проверка исходных данных с помощью цикла с постусловием 6](#_Toc116916038)

[3.4 Проверка результатов расчетов с помощью цикла с постусловием 7](#_Toc116916039)

[3.5 Использование Try … Except … End; 7](#_Toc116916040)

[4 Результаты расчетов 8](#_Toc116916041)

[Приложение А 9](#_Toc116916042)

[Приложение Б 13](#_Toc116916043)

# Постановка задачи

## Исходное условие задачи

Есть определенное количество зеленых бактерий и определенное количество тактов времени, где каждый такт времени зеленые бактерии делятся на зеленые и красные, а красные – превращаются в зеленые. Вывести общее количество бактерий через заданное количество тактов времени.

## Модифицированное условие задачи

Есть определенное количество зеленых и красных бактерий и определенное количество тактов времени, где каждый такт времени зеленые бактерии делятся на зеленые и красные, а красные – превращаются в зеленые. Вывести изначальные значения зеленых и красных бактерий, общее количество бактерий, зеленых и красных бактерий через заданное количество тактов времени.

## Исходные данные

Программа получает от пользователя количества бактерий зеленого и красного цветов, количество тактов времени.

Вывод данных происходит в следующем виде:

Количество зеленых: SumGreen

Количество красных: SumRed

После N тактов времени из Red красных и Green зеленых бактерий образовалось SumAll бактерий в сумме

Если при расчетах количество бактерий стало отрицательным (произошло переполнение типа Integer) пользователь получает следующее сообщение:

Произошло переполнение, попробуйте значения меньше

Последние положительные значения (', (I-1),' такте):');

Количество зеленых: SumRed

Количество красных: (SumGreen - SumRed)

# Структура данных

Таблица 1 – Данные

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| Red | Integer | Количество красных бактерий |
| Green | Integer | Количество зеленых бактерий |
| SumGreen | Integer | Сумма зеленых бактерий |
| SumRed | Integer | Сумма красных бактерий |
| SumAll | Integer | Сумма всех бактерий через N тактов времени |
| N | Integer | Количество тактов времени |
| I | Integer | Вспомогательная переменная для подсчетов |
| L | Boolean | Переменная для проверки условия на ввод неверного типа данных |
| P | Boolean | Переменная для проверки условия на случай переполнения |

# Детали и методика решения задачи

## Детали задачи

Детали задачи таковы:

* начальное значение тактов времени может быть только целым числом больше 0 и, желательно, меньше 40 при больших значениях количества бактерий;
* значения зеленых и красных бактерий могут быть больше 0 или нулевыми.

## Используемые формулы

Проанализировав условия задачи, можно сделать вывод, что сумма бактерий из веток красных и зеленых бактерий изменяется как сумма чисел Фибоначчи. Из чего можно составить такие рекуррентные отношения как:

Таблица 2 – Количество бактерий, если изначально есть 1 зеленая

|  |  |  |
| --- | --- | --- |
| № такта | Количество зеленых бактерий | Количество красных бактерий |
| 0 | 1 | 0 |
| 1 | 1 | 1 |
| 2 | 2 | 1 |
| 3 | 3 | 2 |
| 4 | 5 | 3 |
| 5 | 8 | 5 |

Таким образом, если изначально нам даны только зеленые бактерии, то их количество на шаге n можно найти из суммы количества бактерий двух цветов на шаге n-1:

А количество красных бактерий будет равно следующему:

Таблица 3 – Количество бактерий, если изначально есть 1 красная

|  |  |  |
| --- | --- | --- |
| № такта | Количество зеленых бактерий | Количество красных бактерий |
| 0 | 0 | 1 |
| 1 | 1 | 0 |
| 2 | 1 | 1 |
| 3 | 2 | 1 |

Продолжение таблицы 3

|  |  |  |
| --- | --- | --- |
| № такта | Количество зеленых бактерий | Количество красных бактерий |
| 4 | 3 | 2 |
| 5 | 5 | 3 |

Так как после первого такта все красные бактерии превратились в зеленые, то, начиная со второго такта, количество бактерий каждого цвета будет считаться так же, как и при изначальном наличии лишь зеленых бактерий. Таким образом, если изначально нам даны только красные бактерии, то сперва присваиваем количеству зеленых бактерий число красных:

и расчеты проводим начиная со второго такта времени.

## Проверка исходных данных с помощью цикла с постусловием

При проверке исходных данных использовался цикл с постусловием, так как в таком случае цикл будет запущен хотя бы один раз:

Repeat

Try

Write('Введите кол-во бактерий красного и зеленого');

Writeln(' цветов соответственно: ');

Readln(Red, Green);

While (Red <= -1) or (Green <= -1) do

Begin

Write('Введите кол-во бактерий красного и ');

Writeln('зеленого цветов соответственно: ');

Readln(Red, Green);

End;

Writeln('Введите кол-во тактов времени (до 40): ');

Readln(N);

While (N < 0) do

Begin

Writeln('Введите кол-во тактов времени (до 40): ');

Readln(N);

End;

L := True;

Except

On e:EInOutError do

Begin

Writeln('Введены некорректные данные!');

Writeln('');

L := False;

End;

End;

Until L;

С помощью этого цикла мы повторяем вывод фразы “ Введите кол-во бактерий красного и зеленого цветов соответственно: ” , повторяем получение данных от пользователя в переменные Red и Green, если было правильно введено значение, иначе фразу “Введены некорректные данные! ” до тех пор, пока полученные значения не будет выполнять всем требованиям. Так же происходит и с вводом количества тактов времени.

## Проверка результатов расчетов с помощью цикла с постусловием

При проверке результатов расчетов использовался цикл с постусловием, так как в таком случае цикл будет запущен хотя бы один раз. При проведении расчетов с большими значениями исходных данных происходит переполнение типа Integer, что приводит к отрицательным значениям. В таком случае пользователю выводится фраза «Произошло переполнение, попробуйте значения меньше», такт, когда переполнения не было, значения на нем и программа запускается заново. В противном случае пользователю выводятся результаты расчетов на заданном такте времени.

## Использование Try … Except … End;

В данной задаче данные вводятся пользователем с клавиатуры. Нельзя утверждать, что пользователь введёт верные данные, а именно тип Integer.

Для этого используется «Try … Except … End;» При введении некорректных данных (это может быть как и слишком большое число, так и не число вовсе) происходит ошибка, которую считывает Except и выполняет следующие строки:

Writeln('Введены неверные данные!');

Writeln('');

E := False;

# Результаты расчетов

Вследствие выполнения программы на экран выводятся следующие результаты расчетов:

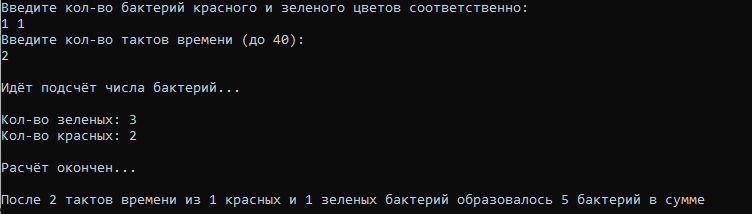


Рисунок 1 – Результаты расчётов

Приложение А

(обязательное)

Исходный код программы

program Бактерии;

// Calculating sum of green and red bacteria after N

// time cycles

Uses System.SysUtils;

var Red, Green, SumGreen, SumRed, SumAll, N, I : integer;

L, P : boolean;

// Red, Green - amount of red and green bacteria

// SumGreen, SumRed - sum of green and red bacteria

// SumAll - sum of green and red bacteria together

// N - amount of time cycles

// I - auxiliary variable

// L, P - checking value

begin

// For case of overflow

Repeat

// Requesting and obtaining the initial amount of

// bacteria and time cycles

Repeat

Try

// Rerequesting and reobtaining the initial

// amount of bacteria

Write('Введите кол-во бактерий красного и ');

Writeln(' зеленого цветов соответственно: ');

Readln(Red, Green);

// if amount isn't right

While (Red <= -1) or (Green <= -1) do

Begin

Write('Введите кол-во бактерий красного и ');

Writeln(' зеленого цветов соответственно: ');

Readln(Red, Green);

End;

// Rerequesting and reobtaining the initial

// amount of time cycles

Writeln('Введите кол-во тактов времени(до 40):');

Readln(N);

// if amount isn't right

While (N < 0) do

Begin

Writeln('Введите кол-во тактов времени(до 40):');

Readln(N);

End;

// Except repeat

L:= True;

// if the user gave the wrong type

Except

On E:EInOutError do

begin

Writeln('Введены некорректные данные!');

Writeln('');

// Go to repeat

L := False;

End;

End;

Until L;

Writeln('');

Writeln('Идёт подсчёт числа бактерий... ');

Writeln('');

SumGreen := Green;

SumRed := Red;

I := 0;

// If we have only red bacteria

if Green = 0 then

Begin

// Conduction the first-time cycle

SumGreen := Red;

SumRed := 0;

// Here we conducted the first-time cycle

// It's necessary to diminish the value of time

// cycles by one

N := N-1;

End;

// Checking for overflow of Integer type and

// implementation dividing

// amount of time cycles of bacteria under given

while (SumGreen>=0) and (SumRed>=0)and (SumAll>=0)and

(I<=N) do

Begin

SumGreen := SumGreen + SumRed;

SumRed := SumGreen - SumRed;

SumAll := SumGreen + SumRed;

// Increasing the value of time cycle

I := I + 1;

End;

// Values output

if (SumGreen>=0) and (SumRed>=0) and (SumAll>=0)then

Begin

Writeln('Количество зеленых: ', SumGreen);

Writeln('Количество красных: ', SumRed);

Writeln('');

Writeln('Расчёт окончен...');

Writeln('');

Write('После ', N, ' тактов времени из ', Red);

Write(' красных и ', Green, ' зеленых бактерий ');

Writeln('образовалось ', SumAll, ' бактерий в

сумме');

// The overflow isn't here

P:= True;

End

else

Begin

Writeln('Произошло переполнение, попробуйте значения

меньше');

Writeln('');

Writeln('Последние положительные значения:');

Writeln('Количество зеленых: ', SumRed);

Writeln('Количество красных: ', (SumGreen - SumRed));

// The overflow is here

P:= False;

End;

Writeln('');

Until P;

readln;

end.

Приложение Б

(обязательное)

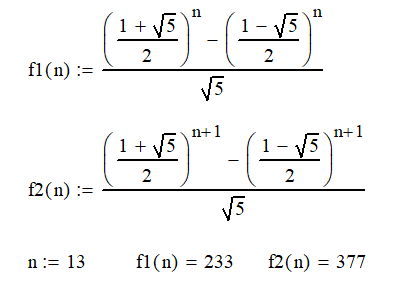
Тестовые наборы

Тест 1

Исходные данные:

* зеленые бактерии – 1;
* красные бактерии – 0;
* такты времени – 13.

Ожидаемый результат:



Полученный результат:

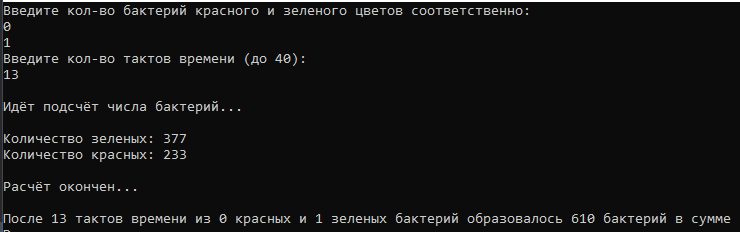


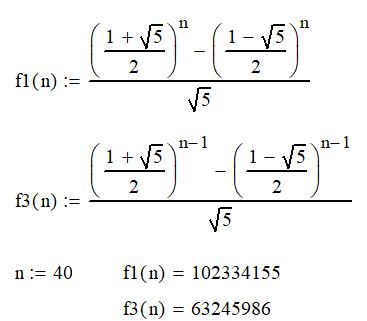
Рисунок 2 – Подсчет результатов при введении количества только зеленых бактерий

Тест 2

Исходные данные:

* зеленые бактерии – 0;
* красные бактерии – 1;
* такты времени – 40.

Ожидаемый результат:



Полученный результат:

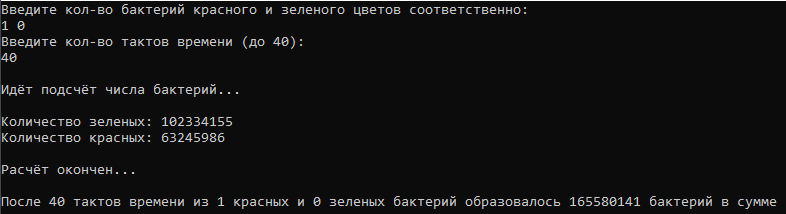


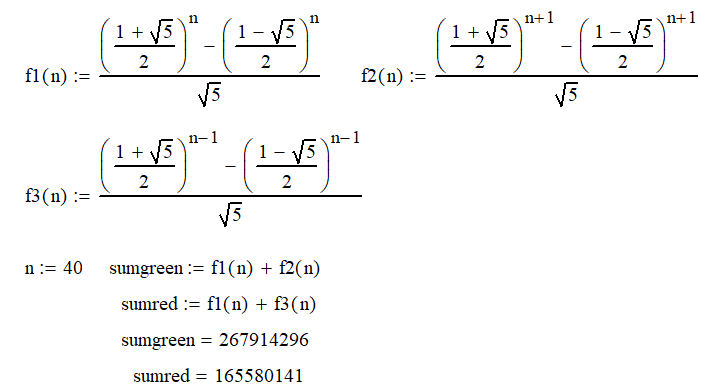
Рисунок 3 – Подсчет результатов при введении количества только красных бактерий

Тест 3

Исходные данные:

* зеленые бактерии – 1;
* красные бактерии – 1;
* такты времени – 40.

Ожидаемый результат:



Полученный результат:

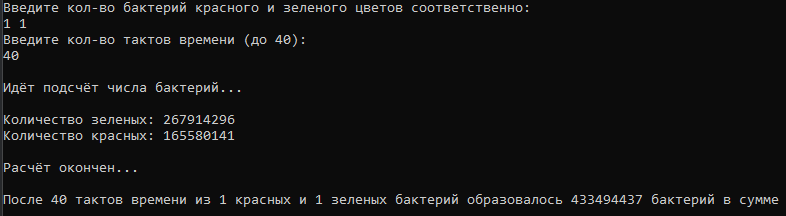


Рисунок 4 – Подсчет результатов при введении количества зеленых и красных бактерий

Тест 4

Таблица 5 – Тестовые ситуации

|  |  |  |
| --- | --- | --- |
| Тест | Тестовые ситуации (вместе с расчётами) | Проверка полученного результата |
| 1. |  | [См. Детали задачи](#_Детали_задачи) |
| 2. |  | [См. Детали задачи](#_Детали_задачи) |
| 3. |  | [См. Детали задачи](#_Детали_задачи) |