МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное

образовательное учреждение

высшего образования

ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ И ФИЗИКО-МАТЕМАТИЧЕСКИХ НАУК

КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

**по дисциплине**

**“ Алгоритмы построение и анализ”**

Выполнил: студент гр. ФИб-3302-51-00 ё. \_\_\_\_\_\_\_\_\_\_\_

Проверил: доцент кафедры ПМиИ Разова Е. В. \_\_\_\_\_\_\_\_\_\_\_

Киров 2020

**Задание 1.**

**Задача 1. Слово Фибоначчи**

**Строками Фибоначчи** (англ. *Fibostring*) называются строки над алфавитом Σ={x,y}, полученные последовательным применением морфизма h:

* h(x)=xy
* h(y)=x

к строке s=y, т. е. последовательность fn(x,y)=hn(y).

**Примеры**:

* f0=y
* f1=x
* f2=xy
* f3=xyx
* f4=xyxxy
* f5=xyxxyxyx

*Лемма 1*. Строки Фибоначчи удовлетворяют рекуррентному соотношению fn=fn−1fn−2, n⩾2.

*Лемма 2*. Для любого целого k⩾0 выполняется fn=fn−k(fk+1,fk).

Например: f7=f5(f3,f2)=(xyx)(xy)(xyx)(xyx)(xy)(xyx)(xy)(xyx).

*Лемма 3*. Для любого целого n⩾2 выполняется равенство fn2=fn+1fn−2.

*Лемма 4*. Для любого целого n⩾3 строка fn имеет грани fi для i=n−2,n−4,…,2−(n mod 2).

**Обратный морфизм** h−1определяется как отображение:

* h−1(xy)=x,
* h−1(x)=

{y,       xx−x,*иначе*y,       xx-x,иначе

Здесь

xx−xx-

 означает, что после этого вхождения x в строке опять следует x.

Обратный морфизм позволяет из строки fnполучить строку fn−1.

**Пример**:

f4=xyxxy.

Будем последовательно применять морфизм:

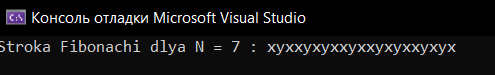
Префикс xy переходит в x, центральный x переходит в y, а суффикс xy также переходит в x.

Получили xyx=f3.

**Задание:**

1. Написать программу генерации строки fn по заданному n.
2. Написать программу генерации по строке fn строки fn-1.

1.1



#include <iostream>

#include <string>

using namespace std;

string fibonachi(int n)

{

string x = "x", y = "y", tmp;

for (int i = 2; i <= n; i++)

{

tmp = x;

x += y;

y = tmp;

}

return x;

}

int main()

{

int n = 7;

cout << "Stroka Fibonachi dlya N = " << n << " : " << fibonachi(n) << endl;

return 0;

}

**Задача 2. Слово Туэ-Морса**

Определим последовательность строк Tn над двухбуквенным алфавитом {a,b} следующим образом: Tn=t0t1…t2n−1, где:

* ti=a, если двоичная запись числа i содержит чётное число единиц
* ti=b, иначе.

Строки этой последовательности называются **строками Туэ-Морса** (англ. *Thue–Morse sequence*).

**Примеры**:

* T0=a
* T1=ab
* T2=abba
* T3=abbabaab
* T4=abbabaabbaababba

**Свойство о получении следующей строки**

Как видно из определения, символ на i-ой позиции не зависит от номера строки. Так как длина строк возрастает, каждая строка является собственным префиксом следующей, поэтому можно рассматривать получение следующей строки как приписывание к текущей строке некоторой другой строки.

Теорема: Пусть φ — морфизм, инвертирующий символы:

φ(x)={b, x=aa,x=b,𝜑x=b, x=aa,x=b,

тогда для строк Туэ-Морса верно следующее соотношение: Tn+1=Tnφ(Tn)

Данная теорема позволяет определять последовательность строк    
Туэ-Морса следующим образом: T0=a, Tn+1=Tnφ(Tn).

1. **Задание:**Написать программу генерации строки Tn по заданному n на основе   
   а) определения;

б) свойства получения следующей строки.



#include <iostream>

#include <string>

using namespace std;

// вернуть дополнение строки

string complement(string s)

{

string comps;

// найти дополнение строки

for (int i = 0; i < s.length(); i++) {

// если 0, добавить 1

if (s.at(i) == '0')

comps += '1';

// если 1, добавить 0

else

comps += '0';

}

return comps;

}

//вернуть n номер последовательности

string nthTerm(int n)

{

string s = "0";

for (int i = 1; i < n; i++)

// добавление добавления к строки

s += complement(s);

return s;

}

int main()

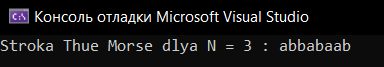
{

int n = 4;

cout << nthTerm(n) << endl;

return 0;

}



#include <iostream>

#include <string>

using namespace std;

// Вернуть дополнение к двоичной строке.

string invert(string s)

{

string invert\_string;

for (int i = 0; i < s.length(); i++) {

if (s.at(i) == 'a') invert\_string += 'b';

else invert\_string += 'a';

}

return invert\_string;

}

string thuemorse(int n)

{

string s = "a";

for (int i = 0; i < n; i++)

s += invert(s);

return s;

}

int main()

{

int n = 3;

cout << "Stroka Thue Morse dlya N = " << n << " : " << thuemorse(n) << endl;

return 0;

}