**Метод на абат Тритемиус**

1. **Класификация**

Методът на абат Тритемиус спада към симетричните криптографски алгоритми. Те се отличават с това, че използват един и същи секретен ключ за процедурите шифроване и дешифриране. Те формират първата голяма група съвременни алгоритми. Силни схеми за криптографска защита могат да бъдат изградени на базата на един от двата основни метода за шифроване на данните – блоково и поточно шифроване. Методът на абат Тритемиус е част от блоковото. Той също така спада и към субституционните алгоритми и може да бъде намерен под името многоазбучно заместване. Групите, на които принадлежи този метод, се отличават с висока скорост на обработка, широко разпространение, удобство за реализация с апаратни и софтуерни средства.

1. **Същност**

Програмата най-общо казано е разделена на две части – шифроване и дешифроване. Още в началото на потребителя му се предлагат двете опции.

1. **Шифроване** – при натискане на „1“, потребителят избира опцията шифроване. След това той трябва да напише съобщението, което желае да криптира и накрая да въведе и ключ. Ключът може да е само една дума или дори няколко. Стъпките в кода следват следната логика:

* Съобщението и ключът се нормализират, т.е. конвертират се в малки букви и се махат празните пространства. След това те се подават на функция, която от стрингови променливи ги превръща в масив от чарове с цел по-лесната им обработка;
* Ключът и дължината на съобщението се подават на функция, която връща масив от чарове, допълнен с дублирания ключ до достигане дължината на съобщението;
* В цикъл за всяка буква от съобщението се извършват следните операции:
* Буквата на ключа се подава на функция, която връща числото, което отговаря на нея. Въпросната функция връща индекса на буквата от масива плюс единица.
* Буквата на съобщението се подава на функция, която връща числото, което отговаря на нея. Въпросната функция връща индекса на буквата от масива плюс единица.
* Резултатната променлива се увеличава като сборът на предните две букви се подава на функция, която връща буква, отговаряща на сбора. Въпросната функция проверява дали подаденото й число е по-малко или равно на дължината на допустимото множество. Ако това е вярно, то тя ще върне буквата на позиция подаденото число минус единица. В противен случай следва още една проверка – ако позицията целочислено разделена на дължината на допустимото множество е равна на нула, то функцията ще върне последната буква на допустимото множество. Но ако условието не е изпълнено, то функцията ще върне целочисленото деление на позицията с дължината на допустимото множество минус единица.
* След като се обходят всички букви на съобщението в конзолата се изписва шифрованото съобщение;

1. **Дешифроване** – при натискане на „2“, потребителят избира опцията дешифроване. Непосредствено той трябва да въведе съобщението за декриптиране. По идентичен начин на шифроването следва въвеждане на ключ, който отново може да е съставен от една или повече думи. Този ключ обаче трябва да бъде предоставен от този, който е криптирал съобщението, тъй като както стана ясно, методът на абат Тритемиус е симетричен. Стъпките в кода следват следната логика:

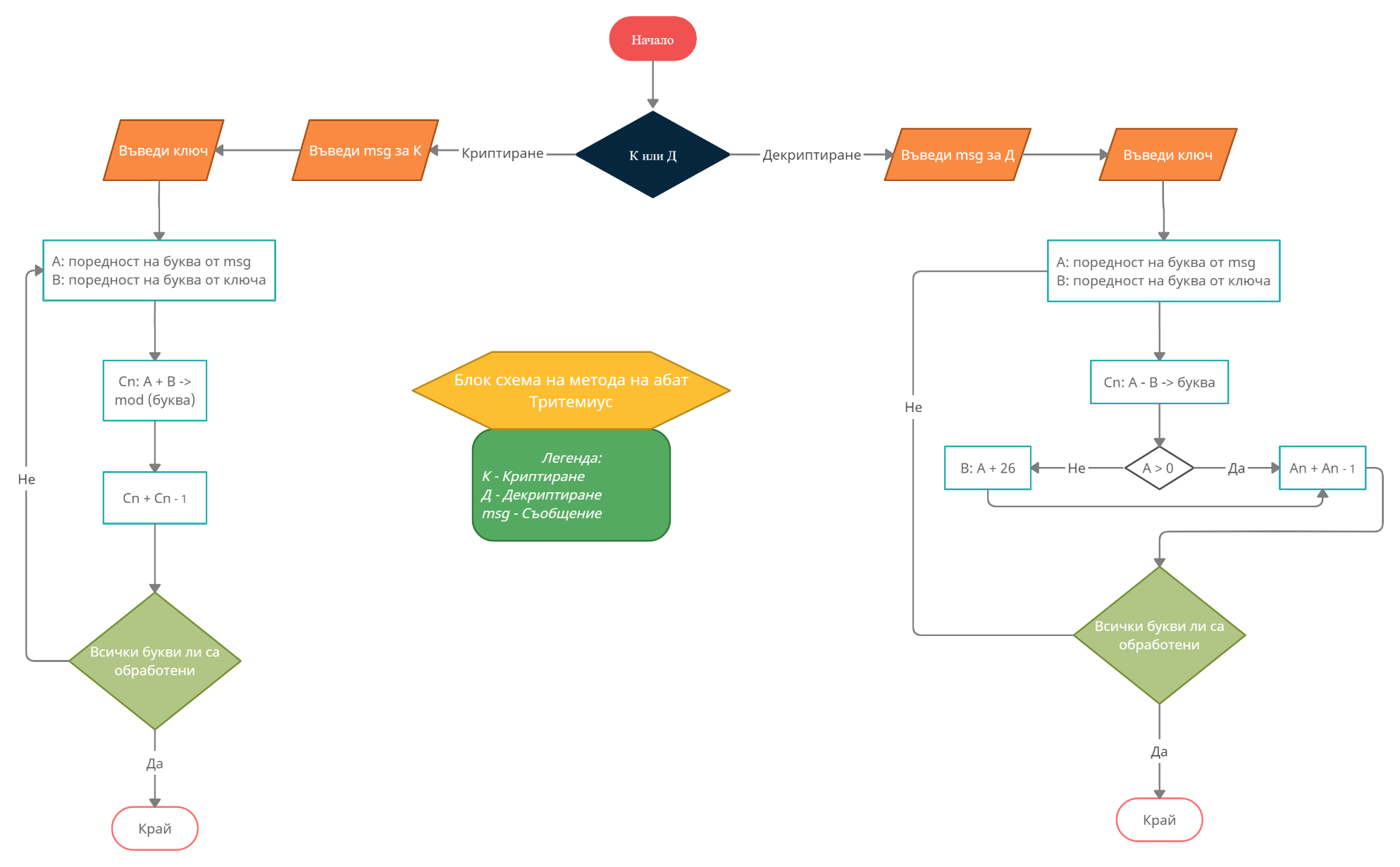
* Съобщението и ключът се нормализират, т.е. конвертират се в малки букви и се махат празните пространства. След това само ключът се подава на функция, която от стрингова променлива я превръща в масив от чарове с цел по-лесна обработка;
* Ключът и дължината на съобщението се подават на функция, която връща масив от чарове, допълнен с дублирания ключ до достигане дължината на съобщението;
* Вика се функция за криптиране, която приема масива от чарове с ключа и съобщението за дешифриране. Тя започва работата си, минавайки през следните стъпки:
* Вика функция за превръщане на стринг в масив от чарове, като подава съобщението за дешифриране.
* За всяка буква от съобщението намира разликата между буквата на съобщението и буквата на ключа. Ако тази разлика е по-голяма от нула, то тя се подава на функцията за връщане на буква от число и тази буква се добавя към крайния резултат. В противен случай към разликата се добавя 26 и след това получения резултат се подава на функцията за връщане на буква от число и върнатата буква се добавя към крайния резултат.
* След като е обходено цялото съобщение, резултатът от дешифроването се изписва в конзолата;

Както може би стана ясно, кодът е разделен на множество функции, така че да е по-лесно четим и в същото време функционален и бърз. По този начин могат лесно да се нанасят промени върху него без да има опасност от непоправими щети.

1. **Блок схема**

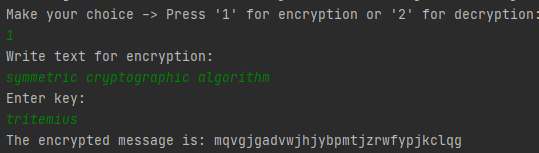
Най-общо казано схемата описва всичко от точка втора схематично. В нея са спазени следните изисквания:

* Елипса: вход или изход
* Квадрат: основно действие
* Успоредни: Вход от конзолата
* Ромб: условие

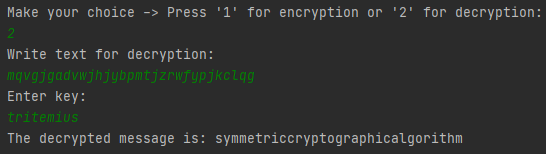
В схемата още са включени коментари по свързочните елементи и легенда за по-лесна четимост.

1. **Резултати от изпълнението на програмата**

Програмата е тествана с множество комбинации. Нагледно са представени три избрани двойки. Първата е с ключ от една дума:

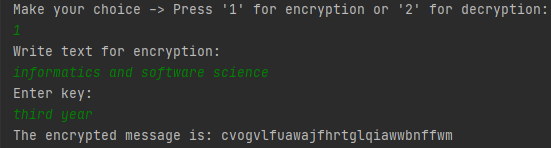


**Шифроване**

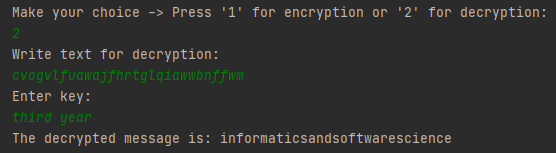
****

**Дешифроване**

Втората е с ключ от две (може и повече):

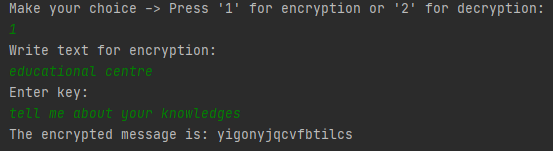
****

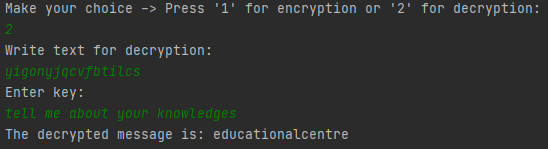
**Шифроване**



**Дешифроване**

Третата е с ключ, по-дълъг от съобщението:





**Шифроване**

**Дешифроване**

Примерите са огледални, за да се провери правдоподобността на алгоритъма.

**Изготвила**: Даяна Димитрова Димитрова **Дата**: 18.10.2021 г.

ФПМИ – ИСН – 77 група