Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Основы алгоритмизации и программирования (ОАиП)

ОТЧЕТ

по разминочной работе №2

Тема работы: Шифрование

Выполнил

студент: гр. 551004 Довыдёнок М.А.

Проверил: Фадеева Е.П.

Минск 2016

Содержание

[1 Постановка задачи 3](#_Toc444983560)

[2 Описание алгоритмов 4](#_Toc444983561)

[3 Результаты расчетов и тестирование программы 5](#_Toc444983562)

[Приложение А 7](#_Toc444983563)

# Постановка задачи

Для шифрование текста, он помещается в квадратную матрицу нечетного размера и считывается по спирали. Реализовать программу, выполняющую шифрование и дешифрование введённого текста. Направлени выбирается случайно и записывается в строке

# Описание алгоритмов

Таблица 1 – Описание алгоритмов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Наименование алгоритма | Назначение алгоритма | Формальные параметры | Рекомендуемый тип |
| 1 | getOperation(result) | Запращивает у пользователя код операции и возвращает его в result | result - boolean. Возвращаемый параметр:  result | Функция. result – результат функции |
| 2 | getStr(result) | Запращивает у пользователя строку result | result - string. Возвращаемый параметр:  result | Функция. result – результат функции |
| 3 | setMatrixLength(matrix, len) | Вычисляет и устанавливает размеры n\*n матрицы matrix | matrix - TMatrix, len – integer. Где TMatrix – array [1..n, 1..n] of char  Возвращаемый параметр: matrix | Процедура |
| 4 | setStrLength(str, len) | Дополняет строку пробелами до необходимой длины | str – string, len – integer. Возвращаемый параметр: str | Процедура |
| 5 | changeDirection(direction) | Устанавливает переменной direction следующее значение | direction – TDirection. Где TDirection – перечеслимый тип из 4 значений. Возвращаемый параметр: direction | Процедура |
| 6 | nextPosition(position, direction) | Вычисляет следующее положение символа, в зависимости от направления direction | position – TPoint, direction – Tdirection. TPoint – запись из 2 полей  Возвращаемый параметр: position | Процедура |
| 7 | strToMatrix(result, operation, str) | Записывает строку в матрицу result  Способ записи зависит от значения operation | result – TMatrix, operation – boolean, str – string. Возвращаемый параметр: result | Функция. result – результат функции |
| 8 | readFromMatrix(result, operation, matrix) | Считывает строку result из матрицы  Способ считывания зависит от значения operation | result – string, operation – boolean, matrix – TMatrix. Возвращаемый параметр: result | Функция. result – результат функции |

# Результаты расчетов и тестирование программы

Тест 1.1 Шифрование

Тестовая ситуация: Шифрование строки из 9 символов

Исходные данные:

Исходная строка: 123456789

Ожидаемый результат:

Полученная строка: 5874123693

Полученный результат:

Рисунок 1 – Тест 1.1

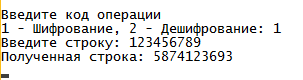


Рисунок 1 – Тест 1.1

Тест 1.2 Шифрование

Тестовая ситуация: Шифрование строки из 25 символов

Исходные данные:

Исходная строка: abcdefghijklmnopqrstuvwxy

Ожидаемый результат:

Полученная строка: mhinsrqlgbcdejotyxwvupkfa1

Полученный результат:

Рисунок 2 – Тест 1.2

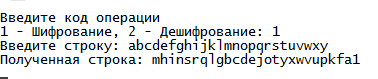


Рисунок 2 – Тест 1.2

Тест 1.3 Шифрование

Тестовая ситуация: Шифрование длинного текста

Исходные данные:

Исходная строка: Lorem ipsum dolor sit amet, consectetur adipisicing

elit. Cupiditate, necessitatibus!

Ожидаемый результат:

Полученная строка: penuingicsubi Cctur a de! t, ieet, condeis

ae.stmdolor sitsilts tttica Lorem ipsum

epiai 3

Полученный результат:

Рисунок 3 – Тест 1.3

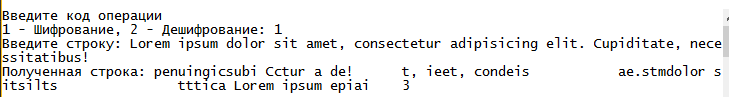


Рисунок 3 – Тест 1.3

Тест 2.1 Дешифрование

Тестовая ситуация: Дешифрование строки из теста 1.1

Исходные данные:

Исходная строка: 5874123693

Ожидаемый результат:

Полученная строка: 123456789

Полученный результат:

Рисунок 4 – Тест 2.1

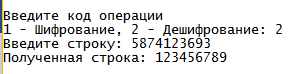


Рисунок 4 – Тест 2.1

Тест 2.2 Дешифрование

Тестовая ситуация: Дешифрование строки из теста 1.2

Исходные данные:

Исходная строка: mhinsrqlgbcdejotyxwvupkfa1

Ожидаемый результат:

Полученная строка: abcdefghijklmnopqrstuvwxy

Полученный результат:

Рисунок 5 – Тест 2.2

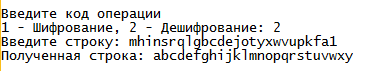


Рисунок 5 – Тест 2.2

Тест 2.3 Дешифрование

Тестовая ситуация: Дешифрование строки из теста 1.2

Исходные данные:

Исходная строка: mhinsrqlgbcdejotyxwvupkfa1

Ожидаемый результат:

Полученная строка: abcdefghijklmnopqrstuvwxy

Полученный результат:

Рисунок 6 – Тест 2.3

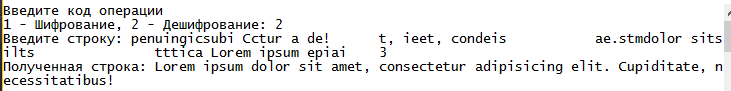


Рисунок 6 – Тест 2.3

Приложение А

(обязательное)

Исходный код программы

program R6;

{$APPTYPE CONSOLE}

uses

Windows;

type

TMatrix = array of array of char;

TDirection = (LEFT, UP, RIGHT, DOWN);

TPoint = record

x, y: Integer;

end;

var

operation: Boolean;

inputStr, resultStr: String;

matrix: TMatrix;

{ Функция, запрашивающая выбор действия }

function getOperation():Boolean;

var

i: integer;

begin

repeat

writeln('Введите код операции');

write('1 - Шифрование, 2 - Дешифрование: ');

readln(i);

until (i = 2) or (i = 1);

if i = 1 then

result := true

else

result := false;

end;

{ Функция, запращивающая ввод строки }

function getStr():String;

begin

write('Введите строку: ');

readln(result);

end;

{ Процедура, находящая и устанавливающая размеры матрицы }

procedure setMatrixLength(var matrix: TMatrix;

const len: Integer);

var

j, i: Integer;

begin

j := 1;

while sqr(j) < len do

inc(j, 2);

setLength(matrix, j);

for i := 0 to j - 1 do

begin

setLength(matrix[i], j);

end;

end;

{ Дополнение строки пробелами }

procedure setStrLength(var str: String; const len: Integer);

begin

while length(str) < len do

str := str + ' ';

end;

{ Изменение направления по часовой стрелке }

procedure changeDirection(var direction: TDirection);

begin

if direction = DOWN then

direction := LEFT

else

direction := succ(direction);

end;

{ Поиск следующей позиции по направлению }

procedure nextPosition(var position: TPoint;

const direction: TDirection);

begin

case direction of

LEFT:

position.x := position.x - 1;

UP:

position.y := position.y - 1;

RIGHT:

position.x := position.x + 1;

DOWN:

position.y := position.y + 1;

end;

end;

{ Заполнение матрицы символами строки }

function strToMatrix(const operation: Boolean;

str: String):TMatrix;

var

i, j, k: Integer;

direction: TDirection;

position: TPoint;

change: Boolean;

begin

case operation of

true:

{ Заполнение по строкам }

begin

setMatrixLength(result, length(str));

setStrLength(str, sqr(length(result)));

k := 1;

for i := 0 to length(result) - 1 do

begin

for j := 0 to length(result) - 1 do

begin

result[i, j] := str[k];

inc(k);

end;

end;

end;

false:

{ Заполнение по обратной спирали }

begin

setMatrixLength(result, length(str)-1);

setStrLength(str, sqr(length(result)));

{ Указание начальной позиции }

position.x := length(result) div 2;

position.y := length(result) div 2;

k := 0;

j := 1;

change := false;

{ Запоминаем направление в последнем символе строки }

direction := TDirection(ord(str[length(str)])-48);

delete(str, length(str), 1);

{ Записываем j символов 2 раза, потом увеличиваем j }

while k < sqr(length(result)) do

begin

i := 1;

while (i <= j) and (k <= sqr(length(result))) do

begin

inc(k);

result[position.y, position.x] := str[k];

nextPosition(position, direction);

inc(i);

end;

changeDirection(direction);

if change then

begin

inc(j);

change := false;

end

else

change := true;

end;

end;

end;

end;

{ Чтение строки из матрицы }

function readFromMatrix(const operation: Boolean;

const matrix: TMatrix):String;

var

i, j, k: Integer;

direction: TDirection;

position: TPoint;

change: Boolean;

ch: Char;

begin

result := '';

case operation of

true:

{ Чтение по спирали }

begin

{ Указание начальной позиции }

position.x := length(matrix) div 2;

position.y := length(matrix) div 2;

k := 0;

j := 1;

change := false;

randomize;

{ Выбор направления }

direction := TDirection(random(4));

ch := chr(ord(direction) + 48);

{ Считываем j символов 2 раза, потом увеличиваем j }

while k < sqr(length(matrix)) do

begin

i := 1;

while (i <= j) and (k <= sqr(length(matrix))) do

begin

result := result + matrix[position.y, position.x];

nextPosition(position, direction);

inc(i);

inc(k);

end;

changeDirection(direction);

if change then

begin

inc(j);

change := false;

end

else

change := true;

end;

{ Добавляем в конец строки символ направления }

result := result + ch;

end;

false:

{ Чтение по строкам }

begin

for i := 0 to length(matrix) - 1 do

begin

for j := 0 to length(matrix) - 1 do

begin

result := result + matrix[i, j];

end;

end;

end;

end;

end;

begin

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

operation := getOperation();

inputStr := getStr();

matrix := strToMatrix(operation, inputStr);

resultStr := readFromMatrix(operation, matrix);

writeln('Полученная строка: ', resultStr);

readln;

end.