Индивидуальное задание

Что предстоит сделать?

Уважаемые студенты! Вам предстоит выполнить творческое задание, результатом которого станет программный продукт, способный решить определенную задачу.

Это может быть компьютерная игра, система шифрования информации и т.д. Обязательное требование, чтобы в данной программе была реализована работа с файлами данных (ввод и вывод).

В данной презентации приведен пример одного из таких заданий, выполненных студентом Артемом Авершиным. Речь идет о создании программы для шифрования данных, а также для оценки его эффективности.

Проверка зависимости корреляции от длины ключа в полиалфавитных шифрах подстановки

Как работать с файлами?

Для работы с файлами необходимо подключить заголовочный файл <fstream>

<ifstream> — файловый ввод, <ofstream> — файловый вывод

- ofstream fout;
- ofstream fout(«test.txt");
- ifstream fin;
- ifstream fin («test.txt");

Открытие файла

- void open (const char* filename, ios_base::openmode mode);

Режимы открытия файлов

- ios_base::in открыть файл для чтения
- ios_base::out открыть файл для записи
- ios_base::ate при открытии переместить указатель в конец файла
- ios_base::app открыть файл для записи в конец файла
- ios_base::trunc удалить содержимое файла, если он существует
- ios_base::binary открытие файла в двоичном режиме

Оператор логического ИЛИ (|) позволяет составить режим с любым сочетанием флагов.

- file.open ("test.txt", ios::out | ios::app);

ofstream: запись файлов

Метод	Описание
• open	Открывает файл для записи
• put	Записывает одиночный символ в файл
• write	Записывает заданное число байт из памяти в файл
• seekp	Перемещает указатель позиционирования в указанное положение
• tellp	Возвращает текущее значение указателя позиционирования файла
• close	Закрывает файл

Запись в файл

Функция put записывает один символ в выходной поток.

- fout.put('A’);
- fout <<'A’;

Функция write записывает блок памяти в выходной файловый поток. Аргумент length указывает количество записанных байт.

- fout.write((char *) &dt, sizeof dt);
- fout << “Запись в файл";

ifstream: чтение файлов

Метод	Описание
• open	Открывает файл для чтения
• get	Читает один или более символов из файла
• getline	Читает символьную строку из текстового файла или данные из бинарного файла до определенного ограничителя
• read	Считывает заданное число байт из файла в память
• eof	Возвращает ненулевое значение (true), когда указатель потока достигает конца файла
• peek	Выдает очередной символ потока, но не выбирает его
• seekg	Перемещает указатель позиционирования файла в заданное положение
• tellg	Возвращает текущее значение указателя позиционирования файла
• close	Закрывает файл

Чтение из файла

Функция getline записывает блок памяти в выходной файловый поток. Аргумент length указывает количество записанных байт.

- fin.getline(buff, 50);
- fin >> buff;

Функции обработки ошибок

Функция	Возвращаемое значение
• bad	Возвращает значение true , если возникла неустранимая ошибка.
• fail	Возвращает значение true , если имеется неустранимая ошибка или ожидаемое условие, например ошибка преобразования или если файл не найден. Обработка часто может возобновиться после вызова clear с нулевым аргументом.
• good	Возвращает значение true , если не удается установить условие ошибки (неустранимое или иное), а флаг конца файла не задан.
• eof	Возвращается true в условии окончания файла.
• clear	Устанавливает внутреннее состояние ошибки. Если вызывается с аргументами по умолчанию, он очищает все биты ошибок.

Закрытие файла

- Функция-член close закрывает файл диска, связанный с потоком выходных файлов

Формулировка задания:

Требуется реализовать алгоритм шифрования Виженера и найти зависимость корреляции исходного и шифрованного сообщения от длины ключа. Корреляция позволит оценить эффективность шифрования.

Линейная корреляция Пирсона

Характеризует существование линейной связи между двумя величинами. Изменяется от –1 до 1.

```
double pearson(char* message1, char* message2, int n) {
    double sx = 0, sy = 0, sxy = 0, sxx = 0, syy = 0, r;
    for (int i = 0; i < n; i++) {
        double x = (double)message1[i];
        double y = (double)message2[i];
        sx += x;
        sy += y;
        sxx += x * x;
        syy += y * y;
        sxy += x * y;
    }

    r = (n * sxy - sx * sy) / sqrt((n * sxx - sx * sx) * (n * syy - sy * sy));
    return r;
}
```

Шифр Виженера

Представляет собой усовершенствованную версию шифра Цезаря с различными значениями сдвига для каждой буквы.

П	Р	О	Г	Р	А	М	М	А	Исходный текст
17	18	16	4	18	1	14	14	1	
К	О	Т	К	О	Т	К	О	Т	Ключ
12	16	20	12	16	20	12	16	20	
29	1	3	16	1	21	26	30	21	Сложение по модулю
Ь	А	В	О	А	Ф	Щ	Э	Ф	Зашифрованный текст

```
char* vigener(char* message, int mLength, char* key, int kLength) {
    char* out = new char[mLength];
    for (int i = 0; i < mLength; i++) {
        out[i] = (message[i] + key[i % kLength] + 1) % 33;
    }
    return out;
}

char* devigener(char* message, int mLength, char* key, int kLength) {
    char* out = new char[mLength];
    for (int i = 0; i < mLength; i++) {
        out[i] = message[i] - key[i % kLength] - 1;
        if (out[i] < 0) out[i] += 33;
    }
    return out;
}
```

Шифр Вернама

Для произведения шифротекста открытый текст объединяется операцией «исключающее ИЛИ» с ключом

П	Р	О	Г	Р	А	М	М	А	Исходный текст
010001	010010	010000	000100	010010	000001	001110	001110	000001	
К	О	Т	К	О	Т	К	О	Т	Ключ
001100	010000	010100	001100	010000	010100	001100	010000	010100	
011101	000010	000100	001000	000010	010101	000010	011110	010101	Исключающее ИЛИ
Ы	Б	Г	Ж	Б	У	Б	Ь	У	Зашифрованный текст

```
char* vernam(char* message, int mlength, char* key, int klength) {
    char* out = new char[mlength];
    for (int i = 0; i < mlength; i++) {
        out[i] = (message[i] ^ key[i % klength]) % 33;
    }
    return out;
}

char* devernam(char* message, int mlength, char* key, int klength) {
    char* out = new char[mlength];
    for (int i = 0; i < mlength; i++) {
        int temp = (!message[i]) ? (message[i] + 32) : (message[i]);
        out[i] = (temp ^ key[i % klength]) % 33 ;
    }
    return out;
}
```

Что реализовано в программе на данный момент?

- Кодирование сообщения шифром Виженера / шифром Вернама
- Декодирование сообщения с использованием ключа
- Сохранение ключей и закодированных сообщений в файл
- Построение таблицы зависимости корреляции от длины ключа

Программный код

```
650  int main() {
651      setlocale(LC_ALL, "Russian");
652      srand(time(NULL));
653
654      char ch = 0;
655      int choise;
656      ifstream fileIn;
657      ifstream fileKey;
658      ofstream fileOut;
659
660      cout << "Выберите действие" << endl;
661      cout << "1. Зашифровать сообщение" << endl;
662      cout << "2. Расшифровать сообщение" << endl;
663      cout << "3. Провести анализ" << endl;
664      cout << "4. Выход" << endl;
665
666      cin >> choise;
667
668      while (cin.fail() || (choise > 4) || (choise < 1)) {
669          cout << "Некорректный ввод данных." << endl;
670          cin.clear();
671          cin.ignore('\0', 32767);
672          cin >> choise;
673      }
674      cin.ignore('\0', 32767);
675      cout << endl;
676
677      switch (choise) {
678      case 1:
679          codeMessage();
680          break;
681      case 2:
682          decodeMessage();
683          break;
684      case 3:
685          analysis();
686          break;
687      case 4:
688          break;
689      }
690
691      system("pause");
692      return 0;
693  }
```

```cpp
#include <iostream>
#include <fstream>
#include <string>
#include <sstream>
#include <cmath>
#include <Windows.h>
#include <iomanip>
#include <conio.h>

using namespace std;

string files[] = { "C:\\Repos\\LR7\\f1.txt",
                   "C:\\Repos\\LR7\\f2.txt",
                   "C:\\Repos\\LR7\\f3.txt",
                   "C:\\Repos\\LR7\\f4.txt",
                   "C:\\Repos\\LR7\\f5.txt",
                   "C:\\Repos\\LR7\\f6.txt",
                   "C:\\Repos\\LR7\\f7.txt",
                   "C:\\Repos\\LR7\\f8.txt",
                   "C:\\Repos\\LR7\\f9.txt",
                   "C:\\Repos\\LR7\\f10.txt" };

string filesKey[] = { "C:\\Repos\\LR7\\fkey1.txt",
                      "C:\\Repos\\LR7\\fkey2.txt",
                      "C:\\Repos\\LR7\\fkey3.txt",
                      "C:\\Repos\\LR7\\fkey4.txt",
                      "C:\\Repos\\LR7\\fkey5.txt" };

 string ciphers[] = { "Шифр Виженера","Шифр Вернама" };
```

```cpp
double pearson(char* message1, char* message2, int n) {
    double sx = 0, sy = 0, sxy = 0, sxx = 0, syy = 0, r;
    for (int i = 0; i < n; i++) {
        double x = (double)message1[i];
        double y = (double)message2[i];
        sx += x;
        sy += y;
        sxx += x * x;
        syy += y * y;
        sxy += x * y;
    }

    r = (n * sxy - sx * sy) / sqrt((n * sxx - sx * sx) * (n * syy - sy * sy));
    return r;
}

char* generateKey(int length) {
    char* key = new char[length];
    for (int i = 0; i < length; i++) {
        key[i] = rand() % 33;
    }
    return key;
}

bool readBit(char num, char bit) {
    return (num & (1 << bit)) == (1 << bit);
}

void writeBit(char* num, char bit, bool val) {
    (val) ? (*num |= (1 << bit)) : (*num &= ~(1 << bit));
}

char* vernam(char* message, int mlength, char* key, int klength) {
    char* out = new char[mlength];
    for (int i = 0; i < mlength; i++) {
        out[i] = (message[i] ^ key[i % klength]) % 33;
    }
    return out;
}
```

```
92  char* devernam(char* message, int mlength, char* key, int klength) {
93      char* out = new char[mlength];
94      for (int i = 0; i < mlength; i++) {
95          int temp = (!message[i]) ? (message[i] + 32) : (message[i]);
96          out[i] = (temp ^ key[i % klength]) % 33 ;
97      }
98      return out;
99  }
100
101 char* vigener(char* message, int mLength, char* key, int kLength) {
102     char* out = new char[mLength];
103     for (int i = 0; i < mLength; i++) {
104         out[i] = (message[i] + key[i % kLength] + 1) % 33;
105     }
106     return out;
107 }
108
109 char* devigener(char* message, int mLength, char* key, int kLength) {
110     char* out = new char[mLength];
111     for (int i = 0; i < mLength; i++) {
112         out[i] = message[i] - key[i % kLength] - 1;
113         if (out[i] < 0) out[i] += 33;
114     }
115     return out;
116 }
117
118 void cyrillic(string* message) {
119     for (int i = 0; i < (*message).length(); i++) {
120         if ((*message)[i] >= -64 && (*message)[i] <= -33) {
121             (*message)[i] += 32;
122         }
123         else if ((*message)[i] == -88) {
124             (*message)[i] = -72;
125         }
126         else if (!((*message)[i] >= -64 && (*message)[i] <= -1) && ((*message)[i] != -72) && ((*message)[i] != -88)) {
127             (*message).erase(i, 1);
128             i--;
129         }
130     }
131 }
```

```
133  char* stringToChar(string* message) {
134      char* arr = new char[message->length()];
135      for (int i = 0; i < message->length(); i++) {
136          if ((*message)[i] >= -26) {
137              arr[i] = (*message)[i] + 33;
138          }
139          else if ((*message)[i] == -72) {
140              arr[i] = 6;
141          }
142          else {
143              arr[i] = (*message)[i] + 32;
144          }
145      }
146      return arr;
147  }
148
149  string charToString(char* message, int length) {
150      string str = "";
151      for (int i = 0; i < length; i++) {
152          if (message[i] > 6) {
153              str += message[i] - 33;
154          }
155          else if (message[i] == 6) {
156              str += -72;
157          }
158          else {
159              str += message[i] - 32;
160          }
161      }
162      return str;
163  }
169  string readFile(ifstream* f) {
170      stringstream ss;
171      string s;
172      while (*f >> s) {
173          ss << s;
174      }
175      ss >> s;
176      (*f).close();
177      return s;
178  }
```

Примеры выполнения программы

1) Шифрование сообщения

```
Выберите действие
1. Зашифровать сообщение
2. Расшифровать сообщение
3. Провести анализ
4. Выход
1

Выберите алгоритм шифрования
1. Шифр Виженера
2. Шифр Вернама
1

Выберите файл с сообщением
1. C:\Repos\LR7\f1.txt
2. C:\Repos\LR7\f2.txt
3. C:\Repos\LR7\f3.txt
4. C:\Repos\LR7\f4.txt
5. C:\Repos\LR7\f5.txt
6. C:\Repos\LR7\f6.txt
7. C:\Repos\LR7\f7.txt
8. C:\Repos\LR7\f8.txt
9. C:\Repos\LR7\f9.txt
10. C:\Repos\LR7\f10.txt
3
```

```
Выберите файл с ключем
1. C:\Repos\LR7\f1.txt
2. C:\Repos\LR7\f2.txt
3. C:\Repos\LR7\f3.txt
4. C:\Repos\LR7\f4.txt
5. C:\Repos\LR7\f5.txt
6. Сгенерировать ключ
6

Выберите способ задания длины ключа
1. В процентах
2. Числом
1

Введите число(от 0 до 100): 75

Выберите файл для сохранения ключа
1. C:\Repos\LR7\f1.txt
2. C:\Repos\LR7\f2.txt
3. C:\Repos\LR7\f3.txt
4. C:\Repos\LR7\f4.txt
5. C:\Repos\LR7\f5.txt
1
Корреляция составила: 0.0283193
```

```
Выберите файл для сохранения сообщения
1. C:\Repos\LR7\f1.txt
2. C:\Repos\LR7\f2.txt
3. C:\Repos\LR7\f3.txt
4. C:\Repos\LR7\f4.txt
5. C:\Repos\LR7\f5.txt
6. C:\Repos\LR7\f6.txt
7. C:\Repos\LR7\f7.txt
8. C:\Repos\LR7\f8.txt
9. C:\Repos\LR7\f9.txt
10. C:\Repos\LR7\f10.txt
10
Для продолжения нажмите любую клавишу . . .
```

2) Расшифровка сообщения

```
Выберите действие
1. Зашифровать сообщение
2. Расшифровать сообщение
3. Провести анализ
4. Выход
2

Выберите алгоритм шифрования
1. Шифр Виженера
2. Шифр Вернама
1

Выберите файл с сообщением
1. C:\Repos\LR7\f1.txt
2. C:\Repos\LR7\f2.txt
3. C:\Repos\LR7\f3.txt
4. C:\Repos\LR7\f4.txt
5. C:\Repos\LR7\f5.txt
6. C:\Repos\LR7\f6.txt
7. C:\Repos\LR7\f7.txt
8. C:\Repos\LR7\f8.txt
9. C:\Repos\LR7\f9.txt
10. C:\Repos\LR7\f10.txt
10
```

```
Выберите файл с ключем
1. C:\Repos\LR7\f1.txt
2. C:\Repos\LR7\f2.txt
3. C:\Repos\LR7\f3.txt
4. C:\Repos\LR7\f4.txt
5. C:\Repos\LR7\f5.txt
1

Выберите файл для сохранения сообщения
1. C:\Repos\LR7\f1.txt
2. C:\Repos\LR7\f2.txt
3. C:\Repos\LR7\f3.txt
4. C:\Repos\LR7\f4.txt
5. C:\Repos\LR7\f5.txt
6. C:\Repos\LR7\f6.txt
7. C:\Repos\LR7\f7.txt
8. C:\Repos\LR7\f8.txt
9. C:\Repos\LR7\f9.txt
10. C:\Repos\LR7\f10.txt
9
Для продолжения нажмите любую клавишу . . .
```

3) Проведение анализа

```
Выберите действие
1. Зашифровать сообщение
2. Расшифровать сообщение
3. Провести анализ
4. Выход
3

Выберите один или несколько алгоритмов шифрования (по одному)
1. Шифр Виженера
2. Шифр Вернама
3. Закончить выбор
1
2
3

Выберите файл с сообщением
1. C:\Repos\LR7\f1.txt
2. C:\Repos\LR7\f2.txt
3. C:\Repos\LR7\f3.txt
4. C:\Repos\LR7\f4.txt
5. C:\Repos\LR7\f5.txt
6. C:\Repos\LR7\f6.txt
7. C:\Repos\LR7\f7.txt
8. C:\Repos\LR7\f8.txt
9. C:\Repos\LR7\f9.txt
10. C:\Repos\LR7\f10.txt
3
```

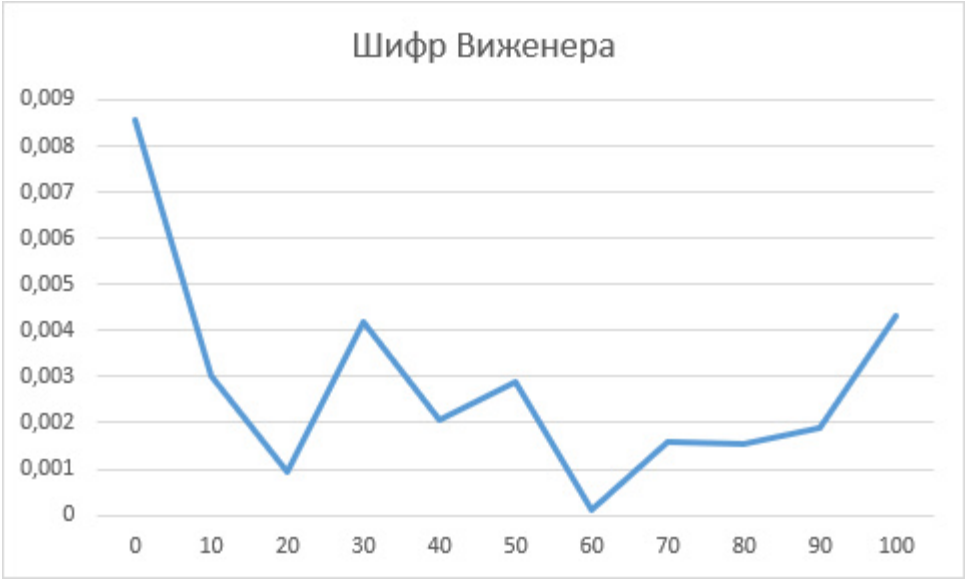
```
Введите шаг изменения ключа(%): 10
Введите количество итераций: 50

Длина ключа(%) Шифр Виженера  Шифр Вернама
           100    0.00429462    0.01481938
            90   -0.00189599    0.01190463
            80   -0.00166074    0.01249046
            70    0.00158356    0.01032859
            60    0.00011573    0.00606904
            50    0.00287200    0.00544483
            40    0.00207124    0.01097818
            30    0.00417327    0.00834931
            20    0.00092341    0.00894416
            10   -0.00301147    0.00518126
             0   -0.00854975    0.02259171
Длина сообщения: 2348 Кол-во итераций: 50
Для продолжения нажмите любую клавишу . . .
```

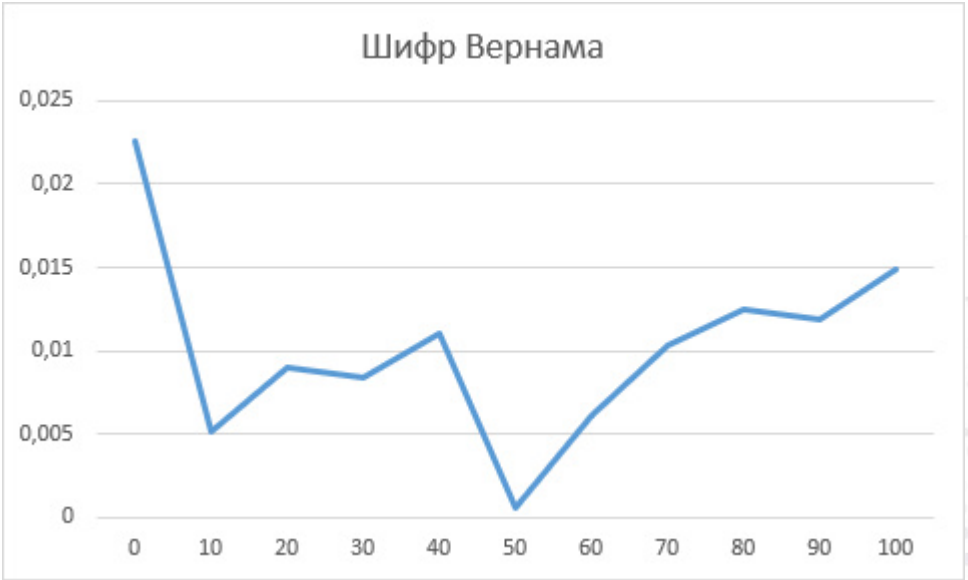
Гипотеза: с приближением длины ключа к длине сообщения корреляция должна достигать своего минимума

Анализ

```
char* generateKey(int length) {
    char* key = new char[length];
    for (int i = 0; i < length; i++) {
        key[i] = rand() % 33;
    }
    return key;
}
```

```
Длина ключа(%) Шифр Виженера  Шифр Вернама
           100    0.00429462    0.01481938
            90   -0.00189599    0.01190463
            80   -0.00166074    0.01249046
            70    0.00158356    0.01032859
            60    0.00011573    0.00606904
            50    0.00287200    0.00544483
            40    0.00207124    0.01097818
            30    0.00417327    0.00834931
            20    0.00092341    0.00894416
            10   -0.00301147    0.00518126
             0   -0.00854975    0.02259171
Длина сообщения: 2348 Кол-во итераций: 50
Для продолжения нажмите любую клавишу . . .
```





Планы на будущее

- Добавить больше шифров
- Реализовать криптографически стойкий генератор псевдослучайных чисел
- Добавить возможность пользователю вводить путь к файлу
- Перейти из консольного приложения в оконное приложение

Спасибо за внимание!