Міністерство освіти і науки України
Національний університет водного господарства
та природокористування

Кафедра обчислювальної техніки

04-04-193

МЕТОДИЧНІ ВКАЗІВКИ

для виконання лабораторних та самостійних робіт
з дисципліни

"Технології захисту інформації"

студентами напряму підготовки
6.050101 "Комп'ютерні науки"

Частина I

Рекомендовано
методичною комісією
напряму підготовки
"Комп'ютерні науки"
Протокол № 6
від 15 березня 2017 р.

Рівне 2017

Методичні вказівки для виконання лабораторних та самостійних робіт з дисципліни "Технології захисту інформації" студентами напряму підготовки 6.050101 "Комп'ютерні науки". Частина І. / П. В. Ольшанський, – Рівне: НУВГП, 2017, – 44 с.

Упорядник П. В. Ольшанський, старший викладач кафедри обчислювальної техніки.

Відповідальний за випуск
Б.Б. Круліковський, кандидат технічних наук, доцент, завідувач кафедри обчислювальної техніки.

Зміст

Лабораторна робота №1

Тема: Реалізація симетричного XOR-алгоритму шифрування файлів в Delphi

Мета роботи. Створити програму для шифрування файлів довільного типу одним з найпоширеніших алгоритмів, відлагодити проект Delphi з декількома модальними формами, спроектувати базовий шаблон для подальшого вивчення різних методів шифрування.

Теоретичні відомості

Прості алгоритми шифрування поділяють на дві групи:

- ***підстановочні***, в яких кожен символ чи група символів заміняється за певним правилом;
- ***перестановочні,*** в яких відбувається перестановки символів.

Підстановочні шифри поділяють на

- ***моноалфавітні*** - для всіх символів використовується однакове правило заміни - елементарні шифри, на практиці не використовуються, бо легко розкриваються з допомогою частотного аналізу зашифрованого тексту;
- ***поліалфавітні*** - послідовні символи тексту заміняються на інші за різними правилами, які визначаються буквами секретного ключа - кожна буква ключа задає певне правило заміни, кількість букв ключа визначає період шифру.
- ***поліграмні*** - групи символів різної довжини заміняються іншими групами з допомогою шифрувальних таблиць або книжок.

Поліалфавітні та поліграмні шифри використовувались багато століть в ручній криптографії до закінчення Другої світової війни,

коли з'явились перші комп'ютери. В сучасних машинних шифрах використовуються в комбінаціях з перестановочними шифрами для надійного перемішування бітів та байтів.

К.Шенноном в 1946р. було доведено, що якщо довжина ключа дорівнює довжині шифрованого тексту і кожен ключ є випадковим набором символів і використовується лише один раз, то такий шифр є абсолютно надійним, і без знання ключа його однозначно неможливо розшифрувати. Однак цей метод, названий методом одноразового блокнота, практично використовується лише в одиничних випадках, бо його неможливо реалізувати для шифрування великих об'ємів інформації.

На практиці використовуються ключі (паролі) обмеженої довжини, які можна запам'ятати чи таємно повідомити користувачам доступними засобами. Найпростішим в реалізації різновидом поліалфавітного шифру є XOR-алгоритм.

Таблиця 1. Логічна операція XOR

A	B	A XOR B
0 (False)	0 (False)	0 (False)
0 (False)	1 (True)	1 (True)
1 (True)	0 (False)	1 (True)
1 (True)	1 (True)	0 (False)

Операція XOR - eXclusive OR, виключне АБО чи додавання за модулем 2 - результат логічної операції з двома двійковими розрядами дорівнює 1, якщо розряди різні, і дорівнює 0, якщо розряди співпадають.

Із вихідного файла побайтово зчитуються дані і з допомогою логічної операції XOR на них послідовно накладаються байти секретного ключа. Зашифровані байти накопичуються в результуючому файлі. Повторне застосування операції XOR відновлює початкове значення, внаслідок чого алгоритм можна використовувати і для шифрування і для розшифровування.

Рис.2 Головна форма проекту

Хід роботи

1. Відкрийте новий проект в Delphi, створіть головну форму і заповніть підписами і кнопками згідно рис.2.

2. Розмістіть на формі компонент OpenDialog1 (закладка Dialogs). У властивості Filter задайте типи файлів згідно рис.3.

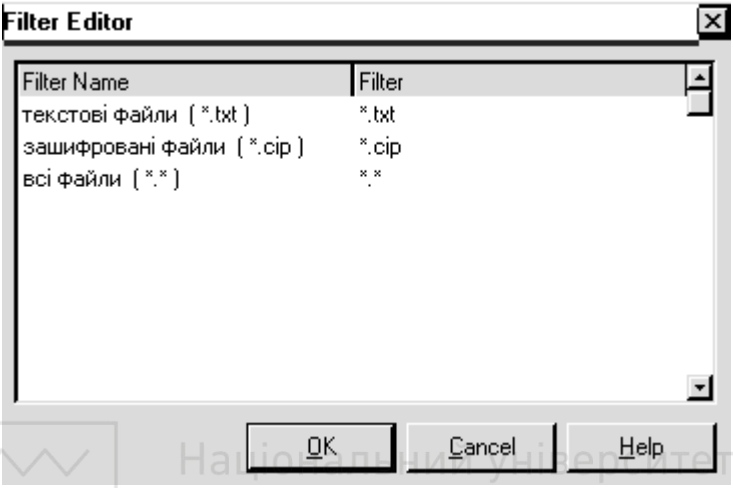

Рис.3 Задання властивостей Filter для OpenDialog1

3. Створіть нову форму Form2 для задавання ключа згідно рис.4. Задайте для фоми властивості: BorderStyle=None; Visible=False; Position=poScreenCenter; Розмістіть на формі компонент Edit. Створіть для форми обробники подій

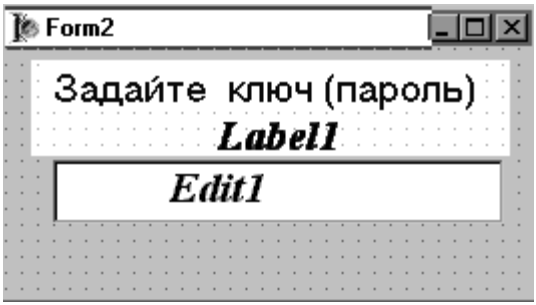

Рис.4 Модальна форма для введення ключа

Рис.5 Форма Form3 для демонстрації процесу шифрування

```
procedure TForm2.Edit1KeyDown(Sender: TObject;
                 var Key: Word;   Shift: TShiftState);
begin
   if (key=VK_RETURN) then
   begin
     psw1:=Edit1.Text;
     ModalResult:=6;
   end;
end;
procedure TForm2.FormShow(Sender: TObject);
begin
 Edit1.Text:='';
end;
```


4. Створіть нову форму Form3 для наочної демонстрації процесу шифрування (рис.5). Задайте властивості BorderStyle=bsDialog; Visible=False; Position=poScreenCenter. Розмістіть на формі п’ять підписів, два поля Memo1 та Memo2, індикатор Gauge1 (закладка Samples).

5. Створіть основну процедуру для шифрування файлів за зразком

```
procedure XOR_alg;
var f,f1:file of byte;
    b,b1:byte;
    s,s1:string;
    k,k100:integer;
    time0,time1:TDateTime;
    FileLen :integer;
```

```
  millisec :comp;
procedure Code(b:byte;var bc:byte);
var   r, a : byte;
begin
  r:=k mod length(psw1);
  a:=ord(psw1[r+1]);
  bc:=a XOR b
end;
begin
   Form3.Caption:='  XOR алгоритм';
   Form3.Label1.Caption:='Вхідний файл:'+name1;
   AssignFile(f,name1);
   Reset(f);  FileLen:= FileSize(f);
   k100:= FileLen div 100;
   Form3.Label3.Caption:='Довжина файла '+
                        IntToStr(FileLen)+'байтів';
   Form3.Label5.Caption:=' ';
   Form3.Label2.Caption:='Зашифрований файл:'+
                                  name1+'.cip';
   AssignFile(f1,name1+'.cip' );
   Rewrite(f1);
   k:=0;// лічильник кількості символів
   Form3.Gauge1.Progress:=0; // покажчик індикатора
   time0:=Time;
   //початок формування рядків для виведення на екран
   s:='';s1:='';
   while not eof(f) do
   begin
  // очищення полів Мемо для запобігання переповнення
```

```
  if k mod 20000=0 then
  begin Form3.Memo1.Clear;  Form3.Memo2.Clear end;
  // відлік відсотків на індикаторі
  if k mod k100=0 then
  begin
    Form3.Gauge1.Progress:=Form3.Gauge1.Progress+1;
    // виведення системного часу
    Form3.Label4.Caption:='Час '+TimeToStr(Now);
    Form3.Refresh; // оновлення форми
  end;
  read(f,b); // читання байта з файла
  // формування рядка для виведення на екран
  s:=s+chr(b);      code(b,b1);  // кодування байта
  write(f1,b1); // запис байта в результуючий файл
  // формування рядка для виведення на екран
  s1:=s1+chr(b1);
  //виведення на екран рядків по 80 символів
  if (b=13) or (length(s)>79) then
  begin
    Form3.Memo1.Lines.Add(s);
    Form3.Memo2.Lines.Add(s1);
    s:='';s1:=''  // очищення рядкових буферів
  end;
  inc(k) // відлік байтів
end;
// виведення залишків символів на екран
Form3.Memo1.Lines.Add(s);
Form3.Memo2.Lines.Add(s1);
time1:=Time-time0; // обчислення часового проміжку
```

```
    Form3.Label4.Caption:='Час шифрування '
          +FormatDateTime(' n хв. ss,zz сек.',time1);
millisec:=TimeStampToMsecs(DateTimeToTimeStamp(time))-
        TimeStampToMsecs(DateTimeToTimeStamp(time0));
    Form3.Label5.Caption:='Швидкість '+
IntToStr(Trunc(FileLen*1000/millisec))+' б/сек';
 // закінчення роботи з файлами
    closefile(f);closefile(f1);
    ShowMessage('Збережено у файлі '+name1+'.cip');
end;
```

6. Для виконання процедур та керування формами створити обробник подій для натискання відповідної кнопки головної форми

```
procedure TForm1.Button1Click(Sender: TObject);
begin
 OpenDialog1.Title:='Виберіть файл для шифрування/
                                розшифровування';
 OpenDialog1.Execute;
 if OpenDialog1.FileName='' then exit;
 name1:=OpenDialog1.FileName;
 Form1.Hide;
 Form2.ShowModal;
 if Form2.ModalResult=6 then
 begin
   ShowMessage('Ваш пароль '''+psw1+'''');
   Form2.Close;
 end;
 Form3.Show;
 XOR_alg;
 Form3.Close;
```

```
  Form1.Show
end;
```

7. В описовому розділі головної форми опишіть глобальні змінні

```
var
  Form1: TForm1;
  psw1,  //  ключ шифрування
  name1  //  назва файла, вибраного для шифрування
  :string;
```

8. Відлагодьте програму та збережіть всі файли проекту в окремому каталозі для звіту.

Контрольні запитання.

- *Назвіть основні типи елементарних алгоритмів шифрування.*
- *Чому логічну операцію XOR називають також додаванням за модулем два ?*
- *Які типи файлів Turbo Pascal та Object Pascal Ви знаєте?*
- *Назвіть стандартні процедури для роботи з файлами.*
- *Чому запропонований алгоритм шифрує файли будь-якого типу?*
- *Які методи криптоаналізу можна застосувати для зламування текстів, зашифрованих алгоритмом XOR?*
- *Які стандартні функції Object Pascal використовуються для обчислення середньої швидкості шифрування?*
- *Запропонуйте вдосконалений варіант реалізованого в роботі алгоритму для прискорення шифрування файлів.*

Лабораторна робота №2

Тема: Реалізація перестановочного алгоритму шифрування файлів в Delphi. Комбінування підстановочних та перестановочних алгоритмів

Мета роботи. Створити програму для шифрування файлів довільного типу з допомогою перестановочного алгоритму, інтегрувати алгоритм в створений раніше проект Delphi, розробити власний оригінальний метод шифрування у вигляді довільної комбінації підстановок та перестановок.

Теоретичні відомості

Перестановочні алгоритми шифрування використовувались в практичній криптографії з давніх давен. Більшість сучасних складних криптографічних машинних алгоритмів (DES, ГОСТ-89 та подібні) для кращого перемішування інформації в зашифрованому файлі включають декілька етапів, на яких переставляються місцями окремі біти, байти або групи байтів. Перестановочні алгоритми легко реалізуються в шифрувальній електронній апаратурі, бо для цього досить перекомутувати провідники, по яких передаються сигнали. Ефективна програмна реалізація цих алгоритмів набагато важча, бо перестановки бітів чи груп байтів не входять до основного набору операцій сучасних процесорів. Розробникам нових програмних алгоритмів шифрування рекомендується уникати перестановок, а використовувати лише підстановки та циклічні зсуви. Однак це рекомендації для майбутнього, бо машинні шифри на основі DES добре вивчені, пройшли тривалу практичну перевірку і довели свою надійність і криптостійкість.

Алгоритм, який розглядається в даній роботі, реалізовує перестановки байтів згідно заданого ключа. Вхідний файл розбивається на порції, довжина яких відповідає довжині ключа. Ключ повинен містити різні букви (символи). Першим в результуючий файл записується той байт (символ), номер якого співпадає з номером наймолодшого (розташованого в алфавіті чи кодовій таблиці раніше інших) символа ключа. Наприклад, в якості ключа використовується слово "YOURSELF" Порядок символів в ключі за алфавітом утворює масив номерів **8, 4, 7, 4, 5, 1, 3, 2.**

Згідно з цими номерами будуть формуватись символи результуючої перестановки.

```
procedure Transposition(s:string; var ss:string);
var i:integer;
begin
 ss:=s;//для ініціалізації рядка символів довж. L
 for i:=1 to L do ss[key_number[i]]:=s[i]
end;
```


Потім процес повторюється для наступної порції інформації. Якщо остання порція є неповною (меншою довжини ключа), можливі різні варіанти реалізації:

а) доповнення порції до необхідної довжини;

б) застосування спеціального алгоритму, який використовує частину ключа;

в) дописування неповної останньої порції у результуючий файл без змін.

В даній роботі вибрано найпростіший третій спосіб.

Розшифровування файлів здійснює процедура, обернена до попередньої

```
procedure Reverse(s:string;  var ss:string);
var i:integer;
begin
 ss:=s;//для ініціалізації рядка символів довж. L
 for i:=1 to L do ss[i]:=s[key_number[i]]
end;
```

Хід роботи

1. Відкрийте проект з попередньої лабораторної роботи в Delphi, доповніть кнопками згідно рис.2.

2. Для кнопок Button2 та Button3 створіть спільний обробник події натискання за наступним зразком


```
procedure TForm1.Button2Click(Sender: TObject);
label m1;
var Sen:Tobject; psw:string; i,j:byte;
procedure Transposition_algorithm;
var f,f1:file of char;
      time0,time1:TDateTime;
      FileLen :integer; millisec :comp;
      c:char;    s,ss:string;
      k, k100, i, j, L, i1, num : integer;
      key_number : array[1..255] of integer;
      smemo1,smemo2 : string;
      Last : Boolean;
begin
    Sen:=Sender;  L:=Length(psw1);
    //   формування числового масиву
```

```
//   з номерами- відповідниками букв ключа
 for i:=1 to L do
 begin
   key_number[i]:=1;
   for j:=1 to L do
    if psw1[i]>psw1[j] then Inc(key_number[i])
  end;
  Form3.Caption:=
      'Алгоритм шифрування перестановками';
 Form3.Label1.Caption:='Вхідний файл:'+name1;
 AssignFile(f,name1);
 Reset(f); FileLen:= FileSize(f);
 k100:= FileLen div 100;
 Form3.Label3.Caption:=
'Довжина файла '+IntToStr(FileLen)+'байтів';
 Form3.Label5.Caption:=' ';
 Form3.Label2.Caption:=
          'Зашифрований файл:'+name1+'.cip';
 AssignFile(f1,name1+'.cip' );
 Rewrite(f1);
 time0:=Time;
 k:=0;    // лічильник кількості символів
 Form3.Gauge1.Progress:=0;//покажчик індикатора
 Last:=False; smemo1:=''; smemo2:='';
 repeat //  початок формування блока із L символів
   s:='';
   for i:=1 to L do
   begin
   // перевірка на досягнення кінця файла
```

```
      if eof(f) then
      begin
        Last:= True; Break
      end;
      read(f,c);  // читання символа з файла
      inc(k);     // відлік кількості символів
  // очищення полів Мемо для запобігання переповнення
      if k mod 20000=1 then
      begin Form3.Memo1.Clear; Form3.Memo2.Clear end;
      // відлік відсотків на індикаторі
 if k mod k100=0 then
 begin
  Form3.Gauge1.Progress:=Form3.Gauge1.Progress+1;
  // виведення системного часу
  Form3.Label4.Caption:='Час '+TimeToStr(Now);
  Form3.Refresh; // оновлення форми
 end;
 s:=s+c       // формування блока із L символів
end;
// остання неповна порція записується у вихідний файл
if Last then
       for i:=1 to Length(s) do write(f1,s[i])
else // шифрування перестановкою символів
    //      блоку із L символів
begin
  ss:=s;
  for i1:=1 to L do
  begin
    num:=key_number[i1];
```

```
    if Sen=Button2 then
     ss[num]:=s[i1] //шифрування одного символа
    else  // розшифровування одного символа
    if Sen=Button3 then ss[i1]:=s[num]
  end;
   // запис зашифрованого блока у файл
  for i:=1 to L do write(f1,ss[i]);
end;
  // формування рядків для виведення на екран
  smemo1:=smemo1+s; smemo2:=smemo2+ss;
  if length(smemo1)>60 then
  begin //виведення на екран рядків по 60 символів
    Form3.Memo1.Lines.Add(smemo1);
    Form3.Memo2.Lines.Add(smemo2);
    // очищення рядкових буферів
    smemo1:=''; smemo2:=''
  end
until Last;
time1:=Time-time0; // обчислення часового проміжку
Form3.Label4.Caption:='Час шифрування '+
         FormatDateTime(' n хв. ss,zz сек.',time1);
millisec:=TimeStampToMsecs(DateTimeToTimeStamp(time))-
        TimeStampToMsecs(DateTimeToTimeStamp(time0));
Form3.Label5.Caption:='Швидкість    '+
     IntToStr(Trunc(FileLen*1000/millisec))+' б/сек';
 // закінчення роботи з файлами
 closefile(f);closefile(f1);
 ShowMessage('Збережено у файлі '+name1+'.cip');
end;
```

```
begin
 OpenDialog1.Title:=
     'Виберіть файл для шифрування/ розшифровування';
 OpenDialog1.Execute;
 if OpenDialog1.FileName='' then exit;
 name1:=OpenDialog1.FileName;
 Form1.Hide;
 Form2.ShowModal;
 if Form2.ModalResult=6 then
 begin // вилучення із ключа символів, що повторюються
   psw:='';
   psw:=psw+psw1[1];
   for i:=2 to length(psw1) do
   begin
     for j:=1 to length(psw) do
                 if psw[j]=psw1[i] then goto m1;
     psw:=psw+psw1[i];
m1:end;
   psw1:=psw;
   ShowMessage('Ваш пароль '''+psw1+''''');
   Form2.Close;
 end;
 Form3.Show;
 Transposition_algorithm;
 Form3.Close;
 Form1.Show
end;
```


3. Додайте до головної форми ще дві кнопки - Button4 і Button5 згідно рис.2. Розробіть власний алгоритм шифрування у вигляді

комбінацій підстановок та перестановок, опишіть його детально в словесній формі та запрограмуйте в обробнику натискання кнопок.

4. Збережіть всі файли проекту в окремому каталозі для звіту.

Контрольні запитання.

- *Назвіть основні відмінності в реалізації підстановочних і перестановочних алгоритмів шифрування.*
- *Запишіть формулу для кількість перестановок із N елементів з повтореннями.*
- *Чи змінює перестановка букв частотні характеристики тексту?*
- *Запропонуйте спосіб криптографічної атаки на перестановочний шифр.*
- *Який порядок перестановок називається лексографічним?*
- *Який тип файлів використовується в алгоритмі?*
- *Як реалізувати аналогічний алгоритм шифрування перестановками бітів?*
- *Чому запропонований алгоритм шифрує файли будь-якого типу?*
- *Запропонуйте свій алгоритм шифрування останньої неповної порції тексту.*
- *Як можна вдосконалити реалізований в роботі алгоритм для прискорення шифрування файлів?*

Лабораторна робота №3.

Тема: Реалізація алгоритму шифрування файлів методом одноразового блокнота в Delphi

Мета роботи. Створити програму для шифрування файлів довільного типу класичним абсолютно надійним алгоритмом, інтегрувати алгоритм в створений раніше проект Delphi, відлагодити проект з декількома модальними формами та організувати роботу в програмі з трьома одночасно відкритими файлами.

Теоретичні відомості

К.Шенноном в 1946р. було доведено, що якщо довжина ключа дорівнює довжині шифрованого тексту і кожен ключ є випадковим набором символів і використовується лише один раз, то такий шифр є абсолютно надійним, і без знання ключа його однозначно неможливо розшифрувати. Однак цей метод, названий методом одноразового блокнота, практично використовується лише в одиничних важливих випадках, бо його неможливо реалізувати для шифрування великих об'ємів інформації. У військовій криптографії, дипломатії, розвідці метод використовується для надійної передачі важливих повідомлень чи особливо відповідальних наказів.

В комп'ютерному варіанті реалізації алгоритма шифрування файла методом одноразового блокнота потрібно вказати ключовий файл з довжиною не менше, ніж той, який підлягає шифруванню. Бажано, щоб ключовий файл містив дані випадкового походження. Якщо ключовий файл містить дані, що піддаються статистичному аналізу (наприклад тексти літературних творів, або BMP-файли, які містять велику кількість однакових символів), це значно ослаблює

ефективність методу. В цьому випадку імовірність криптоаналітичного розкриття шифру зростає до 1/2.

Хід роботи

1. Відкрийте проект з попередньої лабораторної роботи в Delphi, доповніть кнопкою Button6 згідно рис.2.

2. Для кнопки Button6 створіть обробник події натискання за наступним зразком

```
procedure TForm1.Button6Click(Sender: TObject);
procedure File_key_alg;
var f,f1,f_key:file of byte;
    b,b1,b_key:byte; s,s1:string;
    k,k100:integer; time0,time1:TDateTime;
    FileLen :integer; millisec :comp;
begin
   AssignFile(f_key,name_key);
   Reset(f_key);
   Form3.Caption:='Алгоритм одноразового блокнота';
   Form3.Label1.Caption:='Вхідний файл    : '+name1;
   AssignFile(f,name1);
   Reset(f); FileLen:= FileSize(f);
   k100:= FileLen div 100;
   Form3.Label3.Caption:='Довжина файла
'+IntToStr(FileLen)+'байтів';
   Form3.Label5.Caption:=' ';
   Form3.Label2.Caption:=
                 'Зашифрований файл : '+name1+'.cip';
   AssignFile(f1,name1+'.cip' );
   Rewrite(f1);  k:=0;// лічильник кількості символів
```

```
  Form3.Gauge1.Progress:=0; // покажчик індикатора
  time0:=Time;
  // початок формування рядків для виведення на екран
  s:='';s1:='';
  //перевірка на досягнення кінця вхідного файла
  while not eof(f) do
  begin
 // очищення полів Мемо для запобігання переповнення
    if k mod 20000=0 then
    begin Form3.Memo1.Clear;  Form3.Memo2.Clear end;
    // відлік відсотків на індикаторі
    if k mod k100=0 then
    begin
      Form3.Gauge1.Progress:=Form3.Gauge1.Progress+1;
      // виведення системного часу
      Form3.Label4.Caption:='Час '+TimeToStr(Now);
      Form3.Refresh; // оновлення форми
    end;
    read(f,b); // читання байта з файла
    // формування рядка для виведення на екран
    s:=s+chr(b);
    // перевірка на досягнення кінця ключового файла
    if eof(f_key) then reset(f_key);
    // читання байта з ключового файла
    read(f_key,b_key);
    b1:=b XOR b_key;   // кодування байта
    write(f1,b1); // запис байта в результуючий файл
    // формування рядка для виведення на екран
    s1:=s1+chr(b1);
```

```
      //виведення на екран рядків по 80 символів
      if (b=13) or (length(s)>79) then
      begin
        Form3.Memo1.Lines.Add(s);
        Form3.Memo2.Lines.Add(s1);
        s:='';s1:=''  // очищення рядкових буферів
    end;
    inc(k) // відлік байтів
  end;
  // виведення залишків символів на екран
  Form3.Memo1.Lines.Add(s); Form3.Memo2.Lines.Add(s1);
  time1:=Time-time0; // обчислення часового проміжку
  Form3.Label4.Caption:='Час шифрування '+
        FormatDateTime(' n хв. ss,zz сек.',time1);
  millisec:=TimeStampToMsecs(DateTimeToTimeStamp(time))-
        TimeStampToMsecs(DateTimeToTimeStamp(time0));
  Form3.Label5.Caption:='Швидкість   '+
    IntToStr(Trunc(FileLen*1000/millisec))+' б/сек';
  //закінчення роботи з файлами
  closefile(f_key);closefile(f);closefile(f1);
  ShowMessage('Збережено у файлі '+name1+'.cip')
end;
begin
  OpenDialog1.Title:=
        'Виберіть файл для шифрування/розшифровування';
  OpenDialog1.Execute;
  if OpenDialog1.FileName='' then exit;
  name1:=OpenDialog1.FileName;
  OpenDialog1.Title:='Виберіть ключовий файл';
```

```
  OpenDialog1.Execute;
  if OpenDialog1.FileName='' then exit;
  name_key:=OpenDialog1.FileName;
  Form1.Hide;
  Form3.Show;
  File_key_alg;
  Form3.Hide;
  Form1.Show
end;
```

4. Відлагодьте програму та збережіть всі файли проекту в окремому каталозі для звіту.

Контрольні запитання.

- *Яким буде результат, якщо в якості ключового файла вибрати файл, який підлягає шифруванню?*
- *В комбінації з якими іншими відомими Вам методами можна використовувати розглянутий в роботі алгоритм шифрування?*
- *Які методи криптоаналізу можна застосувати для зламування текстів, зашифрованих розглянутим алгоритмом ?*
- *Які недоліки в реалізації можуть ослабити надійність розглянутого алгоритму шифрування?*
- *Чому розглянутий в роботі алгоритм важко реалізувати на практиці для шифрування великих обсягів інформації?*
- *Чи змінює запропонований алгоритм частотні характеристики тексту?*
- *Який тип файлів використовується в алгоритмі? Чому запропонований алгоритм шифрує файли будь-якого типу?*

- *Як реалізувати аналогічний алгоритм шифрування перестановками з використанням ключового файла?*
- *Як можна оптимізувати реалізований в роботі алгоритм для прискорення шифрування файлів?*

Лабораторна робота №4.

Тема: Реалізація симетричного алгоритму гамування шифрування файлів в Delphi

Мета роботи. Створити програму для шифрування файлів довільного типу одним з найчастіше використовуваних на практиці алгоритмів, створити та випробувати власний генератор псевдовипадкових чисел, інтегрувати програму в спільний пакет з раніше розробленими алгоритмами.

Теоретичні відомості

Для генерації дуже довгих ключів шифрування можна використати генератор псевдовипадкових чисел. Стандартні генератори типу функції Random для шифрування використовувати не рекомендується, бо по-перше, вони легко піддаються криптоаналізу, по-друге, їх реалізації відрізняються в різних версіях компіляторів.

Для створення власного генератора можна використати дуже простий алгоритм лінійного конгруентного методу. Послідовність чисел в діапазоні від ***0*** до ***m-1*** формується за рекурентною формулою

$$x_{k+1}=(a\,x_k+c) \bmod m$$

Д. Кнут в другому томі "Мистецтва програмування" наводить такі рекомендації для вибору парамерів:

1) x_0 -довільне;

2) a задовольняє таким умовам:

а) a - просте число;

б) $a \bmod 8 = 5$;

в) $\sqrt{m} < a < m - \sqrt{m}$

3) c - непарне число, таке що

$$\frac{c}{m} \approx \frac{1}{2} - \frac{1}{6}\sqrt{3} \approx 0.21132$$

Одержана послідовність псевдовипадкових чисел - гама шифру - накладається у вигляді послідовності байтів з допомогою операції XOR на файл, який шифрується. Пароль (ключ) використовується для вибору x_0.

Для 16-розрядних чисел (тип Word в Turbo Pascal) можна вибрати:

Діапазон від 0 до 2^{16}-1 =65535

c=13849

$$\sqrt{m} = 2^8 = 256$$

$$m - \sqrt{m} = 65536 - 256 = 65280$$

$$256 < a < 65280$$

Конкретні значення параметра **a** можна вибрати з допомогою паскаль-програми

```
program Select_a;
var a:word;
function Prime(n:Word):Boolean;
```

```
var i,im:word;
begin
  im:=trunc(sqrt(n));
  for i:=2 to im do
    if n mod i =0 then begin Prime:=False; Exit end;
  Prime:=True
end;
begin
   for a:=257 to 65279 do
      if (Prime(a)) AND (a mod 8 =5) then writeln(a)
end.
```

Одержуємо довгий список можливих значень: 269, 277, 293, 317, 349, 373, 389, 1397, 421, 461, 509, 541, 557, 613, …, 65141, 65173, 65213, 65269.

Для 32-розрядних чисел (тип Cardinal в Delphi Object Pascal) можна вибрати:

Діапазон від 0 до 2^{32}-1 = 4 294 967 295

c=907612489

$$\sqrt{m} = 2^{16} = 65536$$

$$m - \sqrt{m} = 4294967296 - 65536 = 4429901760$$

$$65536 < a < 4429901760$$

Для одержання конкретних значень параметра **a** потрібно модифікувати попередню програму, змінивши тип Word на Cardinal, вибрати для циклу невеликий діапазон можливих значень і відкомпілювати, як консольну (Console Application), в Delphi.

```
program select_a32;
{$APPTYPE CONSOLE}
```

```
uses SysUtils;
var a:Cardinal;
function Prime(n:Cardinal):Boolean;
var i,im:Cardinal;
begin  im:=trunc(sqrt(n));
  for i:=2 to im do
    if n mod i =0 then
    begin Prime:=False;
          Exit
    end;
  Prime:=True
end;
begin
{ TODO -oUser -cConsole Main : Insert code here }
for a:=65537 to 400001 do
  if (Prime(a)) AND (a mod 8 =5) then writeln(a);
readln
end.
```

Хід роботи

1. Відкрийте проект з попередньої лабораторної роботи в Delphi, доповніть кнопкою Button7 згідно рис.2.

2. Для кнопки Button7 створіть обробник події натискання за наступним зразком

```
procedure TForm1.Button7Click(Sender: TObject);
procedure Random_alg;
var f,f1:file of byte;
    b,b1,bHi,bLo:byte;
    s,s1:string; i, k,k100:integer;
```

```
    time0,time1:TDateTime;
     FileLen :integer; millisec :comp;
     x:Word;
begin
//встановлення ГПВЧ в позицію, задану ключем шифрування
   randseed:=0;
   for i:=1 to Length(psw1) do
                          inc(randseed,ord(psw1[i]));
   Form3.Caption:='Алгоритм гамування';
   Form3.Label1.Caption:='Вхідний файл:'+name1;
   AssignFile(f,name1);
   Reset(f);  FileLen:= FileSize(f);
   k100:= FileLen div 100;
   Form3.Label3.Caption:=
         'Довжина файла '+IntToStr(FileLen)+'байтів';
   Form3.Label5.Caption:='';
   Form3.Label2.Caption:=
                  'Зашифрований файл:'+name1+'.cip';
   AssignFile(f1,name1+'.cip' );
   Rewrite(f1);
   k:=0;// лічильник кількості символів
   Form3.Gauge1.Progress:=0; // покажчик індикатора
   time0:=Time;
// початок формування рядків для виведення на екран
   s:='';s1:='';
   while not eof(f) do
   begin
  // очищення полів Мемо для запобігання переповнення
     if k mod 20000=0 then
```

```
 begin Form3.Memo1.Clear;  Form3.Memo2.Clear end;
 // відлік відсотків на індикаторі
if (k mod k100=0) OR (k mod k100=1) then
begin
   Form3.Gauge1.Progress:=Form3.Gauge1.Progress+1;
   // виведення системного часу
   Form3.Label4.Caption:='Час '+TimeToStr(Now);
   Form3.Refresh; // оновлення форми
 end;
 read(f,b); // читання байта з файла
 // формування рядка для виведення на екран
 s:=s+chr(b);
 // генерування двобайтового випадкового числа
 x:=random(65536);
 bHi:=Hi(x);   // старший байт
 bLo:=Lo(x);   // молодший байт
 b1:=b XOR bHi; // кодування старшим байтом
 write(f1,b1); // запис байта в результуючий файл
 // формування рядка для виведення на екран
 s1:=s1+chr(b1);
 inc(k); // відлік байтів
 if not eof(f) then
 begin
   read(f,b); // читання байта з файла
  // формування рядка для виведення на екран
   s:=s+chr(b);
   b1:=b XOR bLo;  // кодування молодшим байтом
   // запис байта в результуючий файл
   write(f1,b1);
```

```
      // формування рядка для виведення на екран
       s1:=s1+chr(b1);
       //виведення на екран рядків по 80 символів
       if (b=13) or (length(s)>79) then
       begin
         Form3.Memo1.Lines.Add(s);
         Form3.Memo2.Lines.Add(s1);
         s:='';s1:=''  // очищення рядкових буферів
       end;
       inc(k) // відлік байтів
     end
   end;
   // виведення залишків символів на екран
   Form3.Memo1.Lines.Add(s); Form3.Memo2.Lines.Add(s1);
   time1:=Time-time0; // обчислення часового проміжку
   Form3.Label4.Caption:='Час шифрування '+
          FormatDateTime(' n хв. ss,zz сек.',time1);
millisec:=TimeStampToMsecs(DateTimeToTimeStamp(time))-
      TimeStampToMsecs(DateTimeToTimeStamp(time0));
   Form3.Label5.Caption:='Швидкість   '+
    IntToStr(Trunc(FileLen*1000/millisec))+' б/сек';
   // закінчення роботи з файлами
   closefile(f);closefile(f1);
   ShowMessage('Збережено у файлі '+name1+'.cip');
end;
begin
 OpenDialog1.Title:=
        'Виберіть файл для шифрування/розшифровування';
 OpenDialog1.Execute;
```

```
 if OpenDialog1.FileName='' then exit;
 name1:=OpenDialog1.FileName;
 Form1.Hide;
 Form2.ShowModal;
 if Form2.ModalResult=6 then
 begin
   ShowMessage('Ваш пароль '''+psw1+''''');
   Form2.Close;
 end;
 Form3.Show;
 Random_alg;
 Form3.Hide;
 Form1.Show
end;
```

3. У вищенаведеній програмі для одержання шифруючої послідовності використовується стандартний генератор Object Pascal

```
     x:=random(65536);
```

Початкова позиція генератора формується з використанням заданого ключа

```
randseed:=0;
for i:=1 to Length(psw1) do
                        inc(randseed,ord(psw1[i]));
```

Внесіть зміни у програму для використання власного генератора псевдовипадкових чисел довжиною 16 бітів (тип Word). Опишіть генератор в довідці до програми.

4. Внесіть зміни у програму для використання власного генератора псевдовипадкових чисел довжиною 32 біти (тип Cardinal). Опишіть генератор в довідці до програми.

5. Внесіть зміни у програму для використання нелінійного генератора псевдовипадкових чисел. Опишіть генератор в довідці до програми.

6. Внесіть зміни у програму для використання декількох генераторів псевдовипадкових чисел, які використовуються послідовно по черзі в процесі шифрування. Опишіть генератор в довідці до програми.

7. Внесіть зміни у програму для використання декількох генераторів псевдовипадкових чисел, які використовуються паралельно для одержання спільного випадкового числа. Опишіть генератор в довідці до програми.

Власні генератори псевдовипадкових чисел оформити у вигляді окремих гарно документованих функцій. Проміжні варіанти програми (процедури `Button7Click`) зберегти на дискеті в текстових файлах з різними назвами для звіту.

Контрольні запитання.

- *Чим псевдовипадкові послідовності відрізняються від випадкових?*
- *Які випадкові величини називаються рівномірно розподіленими? Як перевірити рівномірність розподілу довільної згенерованої послідовності чисел?*
- *В яких областях математики та інформатики використовуються випадкові числа?*

- *Чи змінює гамування частотні характеристики тексту?*
- *Які методи криптоаналізу можна застосувати для зламування текстів, зашифрованих алгоритмом гамування?*
- *Який тип файлів використовується в алгоритмі? Чому запропонований алгоритм шифрує файли будь-якого типу?*
- *Як реалізувати аналогічний алгоритм шифрування гамування з використанням перестановок?*
- *В математичних довідниках наводяться таблиці випадкових чисел. Чи можна використовувати їх для шифрування?*
- *Як можна оптимізувати реалізований в роботі алгоритм для прискорення шифрування файлів?*

Завдання для самостійної роботи

Задача 1. У конструкторському бюро розроблені кодові замки двох типів. Замок кожного типу відкривається за допомогою секретної комбінації з *n* цифр, встановлюваної для кожного замка індивідуально. Секретна комбінація замка першого типу повинна обов'язково містити хоча б один 0, а другого типу - не містити нулів. У всьому іншому замки аналогічні. Який тип замків (в залежності від *n*) варто вибрати головному конструктору і чому?

Задача 2. Мальвіна вирішила подарувати Буратіно на Новий рік один пакетик "Мівіни". Сама Мальвіна свято буде зустрічати в іншому місті, а їй дуже хочеться, щоб Буратіно завчасно не витяг подарунок із коробки. Тому Мальвіна вирішила покласти коробку з подарунком у сейф, що закривається на кодовий замок. Комбінацію із цифр

$x_1x_2...x_{100}$, що відмикає замок, вона вибрала так, що на місці з номером i знаходиться остання цифра (N+i-100)-го числа Фібоначчі. Для підбору потрібного числа N Мальвіна написала комп'ютерну програму, яка обчислює послідовно числа Фібоначчі, і з'ясувала, що за одну секунду на комп'ютері Буратіно вдається обчислити 500 нових чисел. Рівно за 5 діб до Нового року Мальвіна передала Буратіно цю програму, повідомила правило побудови кодової комбінації і значення N=216000000. Чи може Буратіно, взагалі не використовуючи комп'ютер, відкрити сейф до настання Нового року? (Примітка: послідовність Фібоначчі означається так: перші два її елементи рівні 1, а кожен наступний дорівнює сумі двох попередніх.)

Задача 3. Для запуску комп'ютера Гаріку Потеру необхідно ввести пароль, що складається із 12 цифр. Коли Гарік записував цей пароль, він випадково пропустив одну цифру. Скільки комбінацій прийдеться перебрати Гаріку (у найпідлішому випадку), щоб запустити комп'ютер?

Задача 4. Для передачі повідомлення по телеграфу кожна буква представляється у вигляді п'ятизначної комбінації з нулів і одиниць, яка відповідає двійковому запису номера даної букви в алфавіті (нумерація починається з нуля; замість букви Ё використовується Е, тому всього виходить 32 букви). Передача п'ятизначної комбінації виконується по кабелю, що містить п'ять провідників, причому кожен двійковий розряд передається по окремому провіднику. Один із провідників виявився ушкоджений, внаслідок чого на прийомному кінці міг бути отриманий як 0, так і 1 незалежно від переданого

значення. Знайдіть передане слово, якщо був прийнятий текст ВПРЗЙЬ.

1	2	3	4	5	6	7	8
А	**Б**	**В**	**Г**	**Д**	**Е,Ё**	**Ж**	**З**
00000	00001	00010	00011	00100	00101	00110	00111

9	10	11	12	13	14	15	16
И	**Й**	**К**	**Л**	**М**	**Н**	**О**	**П**
01000	01001	01010	01011	01100	01101	01110	01111

17	18	19	20	21	22	23	24
Р	**С**	**Т**	**У**	**Ф**	**Х**	**Ц**	**Ч**
10000	10001	10010	10011	10100	10101	10110	10111

25	26	27	28	29	30	31	32
Ш	**Щ**	**Ъ**	**Ы**	**Ь**	**Э**	**Ю**	**Я**
11000	11001	11010	11011	11100	11101	11110	11111

Розв’язок.

В	**П**	**Р**	**З**	**Й**	**Ь**
Р	Я	А	Ч	Щ	М
К	З	Ш	П	Б	Ф
Ж	Л	Ф	Г	Н	Ш
А	**Н**	**Т**	**Е**	**Л**	**Ю**
Г	О	С	Ж	И	Э
А	**П**	**Р**	**Е**	**Л**	**Ь**

Задача 5. Із пункту Ч. в пункт З. прокладений кабель, що складається з п'яти провідників. Провідники зовні однакові. Польовому Агенту Бібігону необхідно правильно приєднати провідники до двох шифрувальних апаратів, які розташовані на кінцях кабелю, так, щоб перша клема одного апарата була з'єднана провідником з першою клемою другого і т.д. У розпорядженні агента є тільки пробник, що

складається із батарейки і лампочки, і пронумеровані таблички. Складіть алгоритм правильного приєднання провідників так, щоб пройти при цьому найменшу відстань?

Задача 6. Під час шифрування вихідного повідомлення кожен його символ (букву чи пропуск) заміняли однією або двома цифрами, причому однакові символи завжди заміняли однаково. Для зручності запису одержану послідовність цифр розбили на п'ятірки. Знайдіть вихідне повідомлення за заданим шифротекстом:

14936 21803 25872 18125 86808 62163 49258 62581 21432
42921 29325 43204 08781 32545 81127 81208 56212 57270
32508 74986 21478 04349 32547 49874 92128 25866 43246
21472 98325 81443 64725 74947 47293 80254 46125 45246
72749 24802 16202 84781 23812 54381 47214 94321 49254
46125 27149 80347 81832 54524 81276 24438 02580 25446
12545 24672 74924 80216 20284 78123 81258 14436 47492
54349 43818 32520 80448 12580 47268 12434 95806 83252
78685 25476 81442 28178 03498 52572 08525 45818 62920
43802 54180 26248 12749 47478 12381 25868 18144 42647
80852 58047 26812 43495 80852 58144 42678 18621 04547
49

Задача 7. Для шифрування цифрових повідомлень радистка Кет складає двохрядкову таблицю, у верхньому рядку якої виписані цифри 0,..,9 одна за одною, а в нижньому - ці ж цифри в довільному порядку. Потім Кет послідовно кожну цифру вихідного повідомлення заміняє відповідною цифрою в нижньому рядку таблиці. Скільки існує варіантів заповнення нижнього рядка таблиці, таких, що після двох

шифрувань зазначеним способом завжди буде отримане вихідне повідомлення.

Задача 8. Для використання шифру "Прямокутна ґратка" виготовляють з паперового прямокутника трафарет розміром 6x10 кліток. Вирізані клітки вибирають так, що при накладенні трафарету на лист паперу того ж розміру чотирма можливими способами кожна клітинка листа "відкривається" рівно один раз. Перші 15 букв тексту повідомлення вписуються в прорізи трафарету (по одній в кожну), потім трафарет повертається на 180 градусів, вписуються наступні 15 букв, трафарет перевертається "навиворіт" і т.д. Результат шифрування виглядає так:

Е	**Е**	**И**	**С**	**А**	**Т**	**Ш**	**С**	**Я**	**И**
К	**О**	**Р**	**Т**	**Л**	**М**	**О**	**Р**	**Г**	**Е**
Б	**К**	**Б**	**Р**	**А**	**И**	**Н**	**И**	**У**	**А**
О	**Ч**	**К**	**И**	**С**	**Т**	**У**	**П**	**Т**	**Р**
Ы		**Е**	**О**	**О**		**С**	**Р**	**Л**	**Ь**
Н	**З**	**У**	**Ы**	**Ю**		**К**			**И**

Який текст був зашифрований?

Задача 9. Фірма "Поп-лавський Unlimited" видає красивий диплом із написом "Сертифікований фахівець із шоу та поп-бізнесу" кожному, хто успішно пройде тестування, яке складається із чотирьох послідовних турів. У ході кожного туру випробовуваний повинен у кожному із 7 питань вибрати один із двох варіантів відповіді. Тур

вважається успішно пройденим, якщо було дано не менше 6 правильних відповідей. По закінченні кожного туру претенденту повідомляється, пройшов він на наступний тур чи вибув. Відомо, що склад питань у кожному турі не змінюється. Число спроб пройти тестування не обмежене, але починати завжди приходиться з першого туру. Вєрка Сердючка зовсім нічого не розуміє в поп-бізнесі (і не збирається!), але дуже хоче одержати диплом. Як їй треба діяти, щоб реалізувати свою мрію швидше?

Задача 10. Під час шифрування повідомлення кожну його букву заміняли її порядковим номером в алфавіті (А на 1, ..., Я на 33). Потім додавали до нього черговий член арифметичної прогресії і, нарешті, виписували залишок від ділення цієї суми на 100. Вийшло ось що:

9 90 61 48 44 36 0 93 77 75 41 40 8 4 84 73 50 36 18 12 90 60 52 47 26 6 79 71 75 41 16 98 94 77 52 33 30 7 96 82 58 40 33 19 2 83 69 67 33 23 96 92 64 44 42 28 6 91 94 71 36 25 17 6 72 63 64 50 18 3 88 60 43 28 19 96 89 68 75 42 30 6 96 91 66 46 18 14 2 72 62 53 38 3 3 86 64 47 25 12 9 94 73 58 37 22 12 88 79 66 46 18 16 83 80 64 40 21 31 88 88 55 58 52 16 99 89 68 65 41 31 2 93 91 57 46 34 10 0 95 70 63 29 22 92 82 73 58 37 12 7 92 64 47 45 30 95 81 75 79 36 34 17 15 69 66 43 32 18 94 85 83 52

Знайдіть вихідне повідомлення.

Задача 11. У Міжнародному Банку "Євроцукорлізинг" працюють 9 директорів, причому вони не довіряють один одному. Головний сейф банку відкривається в тому, і тільки тому випадку, коли всі його замки відкриті. Для кожного замка можна виготовити необхідне число копій ключа. Яке найменше число замків повинне бути в сейфі, і як

потрібно розподілити ключі між директорами, щоб сейф міг бути відкритий тільки тоді, коли разом зберуться не менше 5 директорів?

Задача 12. У комп'ютерній мережі використовуються паролі, що складаються з цифр. Щоб уникнути розкрадання паролів, на диску їх зберігають у зашифрованому виді. При необхідності використання відбувається розшифровування відповідного пароля. Шифрування паролів відбувалося посимвольно одинаковим алгоритмом. Перша цифра залишалась без змін, а результат шифрування кожної наступної цифри залежав тільки від неї та від попередньої цифри.

Відомий список зашифрованих паролів: 4350299891410, 540462295825, 4373382021476, 4356244895826, 428540012393, 4363481142576, 4356244895408 і два паролі 4391383375128, 4343503132936, що зберігаються в зашифрованому вигляді в цьому списку. Чи можна відновити які-небудь інші паролі? Якщо так, то відновіть їх.

Задача 13. Дільник d натурального числа a назвемо максимальним, якщо жоден дільник числа a, крім самих a і d, не ділиться на d. Наприклад, числа 4, 6 і 12 є максимальними дільниками для 12, а числа 1, 2 і 3 - ні. Спадну послідовність натуральних чисел будемо вважати правильною, якщо кожен її член, починаючи з другого, є максимальним дільником попереднього. Знайдіть кількість різних правильних послідовностей, що починаються числом 226917 і закінчуються одиницею.

Задача 14. Кожну букву повідомлення замінили парою цифр, що відповідають її порядковому номеру в алфавіті (А замінили на 01, Б - на 02, ..., Я - на 33). Послідовність цифр, що *вийшла,* a_1, a_2, ...

поелементно склали з цифровою *послідовністю* b_1, b_2, ..., після чого виписали залишки одержаних сум від ділення на 10. Вийшло ось що:

1000911693803590924503091895982015921197082902801498921610852004 8529

Знайдіть вихідне повідомлення, якщо додатково відомо, що послідовність b_1, b_2, ... відповідає цифрам десяткового представлення частки 9419/16835, взятими підряд, починаючи з деякого номера.

Задача 15. Під час шифрування тексту, записаного в 32-буквеному алфавіті, кожна його буква замінялася буквою того ж алфавіту, причому різні букви замінялися різними (такий шифр називається *простою заміною*). Ключем такого шифру називається таблиця, у якій для кожної букви алфавіту зазначено, якою буквою її треба замінити. Скільки існує різних ключів, таких, що після дворазового шифрування зазначеним способом з цим ключем, повідомлення МАЄМОТЕЩОМАЄМО перейде в себе?

Задача 16. Тато Карло вирішив відправити Буратіно шифровану телеграму. Кожній букві російського алфавіту він присвоїв числове значення: букві А - 0, букві Б -1, :, букві Я - 32. Потім він вибрав "секретне" слово і приписав до нього праворуч текст вихідного повідомлення. Під кожною буквою вихідного повідомлення Карло виписав суму числових значень цієї букви і k попередніх букв вихідного повідомлення, де k - довжина секретного слова. Нарешті, кожна сума замінена буквою, числове значення якої дорівнює залишку від ділення цієї суми на 33. Вийшло ось що:

СТДЖЬШЦМЁККЬЗЧЖСТГВСУГЖПЧЩИЫЬЪЙЮСАЪЧЧТБЬКЙБЛИЛЛСО

Яке повідомлення послав Карло?

Задача 17. У центральному комп'ютері Кострубатої мережі пейджингового зв'язку з'явився вірус. Він перетворює повідомлення так, що всі букви передаються без перекручувань, а кожна цифра декілька разів шифрується способом, описаним в **задачі 7**, причому кількість шифрувань дорівнює абонентському номеру одержувача повідомлення. Крім того, раз в день таблиця шифрування міняється на іншу. Мальвіна хоче вибрати абонентський номер так, щоб не залежати від дій вірусу. Знайдіть всі номери, які її влаштують.

Задача 18. Один з дев'яти директорів банку "Євроцукорлізинг" обраний головою, у зв'язку з чим потрібно внести зміни в конструкцію сейфа Тепер потрібно, щоб могли відкрити сейф не тільки будь-які п'ять директорів, що зібралися разом , але і будь-який директор разом з головою. Група з менше, ніж п'яти директорів, серед яких немає голови, або голова поодинці не повинні мати можливості відкрити сейф. Яке найменше число замків треба встановити на двері сейфа, і яким чином роздати ключі директорам для реалізації зазначених правил доступу до вмісту сейфа?

Задача 19. а) Чи може сума квадратів двох непарних чисел бути квадратом цілого числа?

б) чи може сума квадратів трьох непарних чисел бути квадратом цілого числа?

Задача 20. Доведіть, що сума квадратів п'яти послідовних натуральних чисел не є точним квадратом.

Задача 21. Доведіть, що сума квадратів трьох натуральних чисел, зменшена на 7, не ділиться націло на 8.

Задача 22. Сума трьох натуральних чисел, що є точними квадратами, ділиться націло на 9. Доведіть, що з них можна вибрати два, різниця яких також ділиться на 9.