

Отчет к лабораторной работе №1

Круглов В.А.

Main.pas

```
unit Main;

{$mode objfpc}{$H+}

interface

uses
  Classes, SysUtils, FileUtil, LResources, Forms, Controls, Graphics, Dialogs,
  StdCtrls, Grids, ExtCtrls, ComCtrls;

type
  { TMainForm }

TMainForm = class(TForm)
  btnGenerate: TButton;
  btnNumbers: TButton;
  cbIN: TComboBox;
  cbOUT: TComboBox;
  GroupBox4: TGroupBox;
  Label4: TLabel;
  Label5: TLabel;
  LabeledEdit1: TLabeledEdit;
  LabeledEdit2: TLabeledEdit;
  LabeledEdit3: TLabeledEdit;
  LabeledEdit4: TLabeledEdit;
  LabeledEdit5: TLabeledEdit;
  ListView1: TListView;
  lnCount: TLabeledEdit;
  bnCount: TLabeledEdit;
  GroupBox1: TGroupBox;
  GroupBox2: TGroupBox;
  GroupBox3: TGroupBox;
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  lCount: TLabeledEdit;
  ListBox1: TListBox;
  ListBox2: TListBox;
  bCount: TLabeledEdit;
  procedure btnGenerateClick(Sender: TObject);
  procedure btnNumbersClick(Sender: TObject);
  procedure cbChange(Sender: TObject);
  procedure FormCreate(Sender: TObject);
private
  TableUnsorted: array [0..99] of word;
  TableSorted: array [0..99] of word;
  TableHash: array [0..2,0..130] of word;
  inNumber, outNumber: word;
  procedure InsertHash(value: word; count: byte);
  procedure Sort;
  function SearchCount(value: word): integer;
  function Search(value: word): boolean;
  function Random: word; overload;
  procedure bSearch;
  procedure lSearch;
public
  { public declarations }
end;
```

```

var
  MainForm: TMainForm;

implementation

{ TMainForm }

procedure TMainForm.btnGenerateClick(Sender: TObject);
var i,j:byte;
    value:word;
    item:TListItem;
    count1,count2:longword;
begin
  Randomize;
  for i:=0 to 130 do
    for j:=0 to 2 do
      TableHash[j,i]:=0;
  for i:=0 to 99 do
  begin
    value:=Random;
    while search(value) do
      value:=Random;
    TableUnsorted[i]:=value;
    TableSorted[i]:=value;
    InsertHash(value, i);
  end;
  Sort;
  ListBox1.Clear;
  ListBox2.Clear;
  ListView1.Clear;
  for i:=0 to 99 do
  begin
    ListBox1.Items.Add(IntToStr(TableUnsorted[i]));
    ListBox2.Items.Add(IntToStr(TableSorted[i]));
  end;
  count1:=0;
  for i:=0 to 130 do
  begin
    item:=ListView1.Items.Add;
    if TableHash[0,i]=0 then
    begin
      item.Caption:='empty';
      item.SubItems.Add('0');
      item.SubItems.Add('0');
    end
    else
    begin
      item.Caption:=IntToStr(TableHash[0,i]);
      item.SubItems.Add(IntToStr(TableHash[1,i]));
      item.SubItems.Add(IntToStr(TableHash[2,i]));
    end;
    count1:=count1+TableHash[1,i];
  end;
  LabeledEdit1.Text:=IntToStr(count1);
  i:=0;
  cbIN.Items.Clear;
  cbOUT.Items.Clear;
  count1:=0;
  count2:=0;
  for value:=200 to 400 do
    if TableSorted[i]=value then
    begin
      Inc(i);
      cbIN.Items.Add(IntToStr(value));
      count1:=count1+SearchCount(value);
    end
    else
    begin
      cbOUT.Items.Add(IntToStr(value));
      count2:=count2+SearchCount(value);
    end;
  end;
end;

```

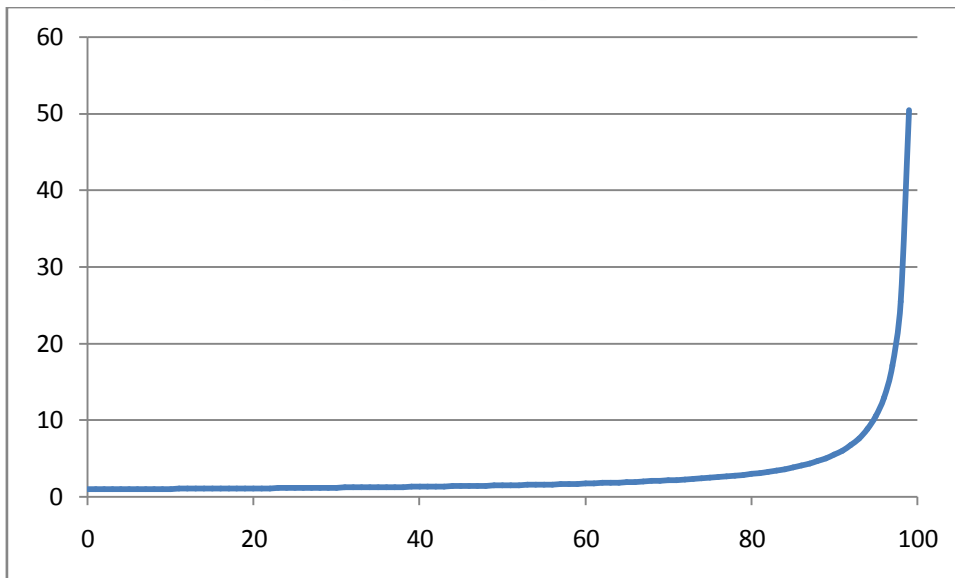
```
end;  
LabeledEdit2.Text:=IntToStr(count1);  
LabeledEdit3.Text:=IntToStr(count2);  
LabeledEdit4.Text:=FloatToStr(count1/100);  
LabeledEdit5.Text:=FloatToStr(count2/100);  
btnNumbersClick(Sender);  
end;
```

Расчет числа проверок

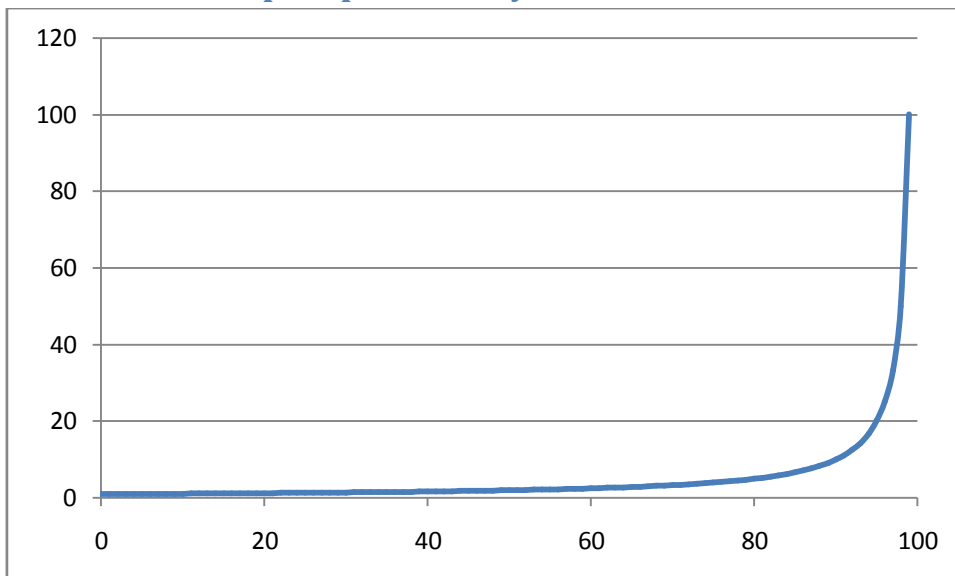
Hash Table

Full access	324
E Tr	324
S Tr	3,24
E Tn	3062
S Tn	30,62

Расчет кол-ва проверок на обнаружение



Расчет кол-ва проверок на отсутствие



Краткие выводы