

```

1: //((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
2: //calculates a cipher suite from string & generates lotto numbers
3: //DEMONSTRATES maXbox with use of INCLUDE & DLL; loc's =182
4: //((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
5:
6:
7: program maXbox_Demonstrator_CODESIGN_Show;
8:
9: function _SayHello2: boolean;
10: external '_SayHello2@income.dll stdcall';
11:
12: const
13:   //something to enlarge the entropy like substitution-box
14:   C1 = 52845;
15:   C2 = 22719;
16:   FNAME = 'readmefirst_maxbox.txt';
17:   CIPHERNAME = 'crypto_readmefirst.txt';
18:   MYKEY = 16549813;
19:   LOTNUMB = 6;
20:   LOTRANGE = 45;
21:
22: type
23:   TLottoList = array[1..LOTNUMB] of byte;
24:   T_Status =(success, failure, pending);
25:   TPasStyle = (a,b,c,d,e,f);
26:   TPasStyles = set of TPasStyle;
27:
28: var
29:   TPas: TPasStyles;
30:   XLApp: Variant;
31:
32:
33: {$I maxdefine.inc}
34:
35: {$I pas_includebox.inc}
36:
37:
38: //-----cryptor service-----//
39:
40: function EncryptDecrypt(const S: string; Key: word; mode: char): string;
41: var
42:   i, myint: word;
43: begin
44:   SetLength(Result, length(S));
45:   //cipher in order of ord ;)
46:   for i:= 1 to Length(S) do begin
47:     myint:= ord(S[i]) xor (Key shr 8)
48:     Result[i]:= chr(myint);
49:     if mode = 'E' then
50:       Key:= (ord(Result[i]) + Key) * C1 + C2;
51:     if mode = 'D' then
52:       Key:= (ord(S[i]) + Key) * C1 + C2;
53:   end;
54: end;
55:
56: procedure StringToFile(acontent, afilename:string);
57: begin
58:   with TFileStream.Create(afilename, fmOpenWrite or fmCreate) do begin
59:     try
60:       Write(acontent, length(acontent));
61:     finally
62:       Free;
63:     end
64:   end
65: end;
66:
67: function FileToString(afilename: string):string;
68: begin
69:   with TFileStream.Create(afilename, fmOpenRead or fmShareCompat) do begin
70:     try
71:       SetLength(result, Size);
72:       Read(result, Size);
73:     finally
74:       Free;
75:     end

```

```

76: end
77: end;
78:
79: function StartLoadService: TStringList;
80: begin
81:   if fileExists(FNAME) then begin
82:     result:= TStringList.create;
83:     result.loadfromfile(FNAME);
84:   end
85: end;
86:
87: //-----lotto generator-----//
88:
89: function initArray(var larray: TLottoList): boolean;
90: var i: byte;
91: begin
92:   for i:= 1 to LOTNUMB do
93:     larray[i]:= 0;
94:     result:= true
95:   end;
96:
97: procedure putNumbers(vrange: byte; var larray: TLottoList);
98: var ti, z, myrand, sum: byte;
99:     dup: boolean;
100: begin
101:   ti:= 1;
102:   sum:= 0;
103:   Randomize;
104:   //precondition set by validation
105:   if vrange < LOTNUMB then vrange:= 6;
106:   repeat
107:     myrand:= Random(vrange)+ 1;
108:     dup:= false;
109:     for z:= 1 to LOTNUMB do
110:       if (larray[z] = myrand) then dup:= true;
111:       if (not dup) then begin
112:         larray[ti]:= myrand;
113:         ti:= ti + 1;
114:         sum:= sum + myrand
115:       end;
116:     //postcondition & proof
117:     until ti = LOTNUMB + 1
118:     writeln('sum of numbers: ' + intToStr(sum))
119:   end;
120:
121: procedure viewNumbers(larray: TLottoList);
122: var i: byte;
123: begin
124:   for i:= 1 to LOTNUMB do
125:     showMessage('lottonumber '+inttostr(i) +': '+inttostr(larray[i]))
126:   end;
127:
128: function startLottoService(lotrange: byte): T_Status;
129: var larray: TLottoList;
130: begin
131:   result:= failure;
132:   if initArray(larray) then begin
133:     putNumbers(LOTRANGE, larray)
134:     viewNumbers(larray);
135:     result:= success
136:   end;
137: end;
138:
139: //-----lotto generator-----//
140:
141:
142:
143: var
144:   mycipher, cipherback: string;
145:   hashback: byte;
146:
147: begin
148:   showMessage('first you have seen the two include files: '#13 #10
149:     + '{${I maxdefine.inc} and {${I pas_includebox.inc}#' #13 #10
150:     + ' then you will see 6 lotto numbers:');

```

```
151:   if startLottoService(lotrange) = success then
152:     writeln('lotto service get at: ' + Now) else
153:     writeln('bad news from service: ' + Now)
154:   showMessage('now a hash from "maxland" a include function: '
155:               +intToStr(hash2('maxland')));
156:   //first encrypt loaded file and store it
157:   showMessage('so now maXbox is to cipher the file '+FNAME);
158:   try
159:     mycipher:= encryptDecrypt(StartLoadService.text, MYKEY, 'E')
160:     StringToFile(mycipher, CIPHERNAME)
161:     //second decrypt file and store it back
162:     cipherback:= FileToString(CIPHERNAME);
163:     if length(mycipher) = length(cipherback) then begin
164:       writeln(encryptDecrypt(cipherback, MYKEY, 'D'))
165:       StringToFile(encryptDecrypt(cipherback, MYKEY, 'D'), FNAME)
166:     end
167:   finally
168:     //fileList.free; is just missing!!
169:     beep2(444, 400)
170:   end
171:   showMessage('then we call a C++ dll function: ')
172:   //from C++ DLL
173:   if _SayHello2 then
174:     writeln('dll invocation realised');
175:   //from include file
176:   showMessage('and at last some include routines: ')
177:   write(floatToStr(intToFloat(234)));
178:   write('run forest run:')
179:   write(intToHex(111123,8))
180:   //write('this one liner')
181:   writeln(AddThousandSeparator('400500210', ''));
182:   showAboutBox;
183: end.
```