

# CrickSpot

Aplicatie Software  
pentru Criptare  
cu Sapte Porti de Protectie

Atestat 2003

Andrei Neculau

Colegiul National Vasile  
Alecsandri, Galati

Profesori Coordonatori:

Prof. Ilie Cristina

Prof. Balacea Georgeta

CrickSpot - Aplicatie Software  
pentru Criptare /P2  
Atestat 2003 - Andrei Neculau  
Colegiul National Vasile  
Alecsandri, Galati

## OBIECTIV PREDEFINIT

---

Obiectivul acestei aplicații software a fost stabilit ca fiind acela de a realiza un cod simplu, un obiect folositor și ușor de utilizat care să cripteze informații în diferite formate (documente, imagini, etc.) într-un număr cât mai mare de pași pentru a oferi securitate maximă, fără a se folosi elemente dinamice.

Proiectul și-a făcut prima apariție acum patru ani, denumit atunci CodeSmart, la concursul din cadrul Săptămânii Colegiului „Vasile Alecsandri”. Din momentul acela, deși a fost premiat cu locul 1, a suferit modificări și îmbunătățiri la nivel de interfață, și bineînțeles la nivel de programare.

CrickSpot este azi foarte ușor de folosit datorit integrării sale în meniurile ®Microsoft Windows, fiind programat similar, ca interfață, după programul de compresie ®WinAce. Deși nu merge după tiparul clasic al conceptelor de criptare, anume aceea de a folosi tabele dinamice, CrickSpot conferă securitate maximă, datorită celor șapte porți de criptare, iar acum, datorită implementării unui concept de compresie denumit Bzip, arhiva rezultată este chiar mai mică decât fișierele inițiale. Pe lângă toate acestea, el implementează conceptul de parolă publică, făcând astfel imposibilă decriptarea conținutului arhivei fără cele două parole prestabilite de către cele două părți participante : expeditorul și destinatarul.

CrickSpot - Aplicatie Software  
pentru Criptare /P3  
Atestat 2003 - Andrei Neculau  
Colegiul National Vasile  
Alecsandri, Galati

## PE SCURT

---

CrickSpot își ia numele pornind de la scopul său, criptare (CRYPT), trecând prin procesul foarte ușor, asemănător unui CLICK, și terminând cu rezultatul foarte mic ca mărime, ce este ca o simplă pată (SPOT).

Pentru a face acest produs utilizabil de cât mai multe persoane interesate, limba proiectului a fost decisă ca fiind limba engleză. De asemenea, au fost folosite componente ®Microsoft Windows care sunt specifice limbii sistemului de operare (în acest caz, limba cea mai răspândită fiind limba engleză).

În principiu, CrickSpot funcționează exact ca un arhivator, un program de compresie, având posibilitatea de a cripta în același timp, cu aceleași parole mai multe fișiere de diferite tipuri. În momentul decriptării, utilizatorul are posibilitatea și de a selecta acele fișiere care îi trebuie, eliminând astfel problema dezarhivării întregului conținut spre a vizualiza doar un singur fișier.

Acțiunea de criptare și decriptare se folosește de facilitatea de multitasking a sistemului de operare ®Microsoft Windows, aceasta desfășurându-se în fundal, fără a bloca interfața grafică a programului. Acesteia i se pot aloca și diferite nivele de prioritate.

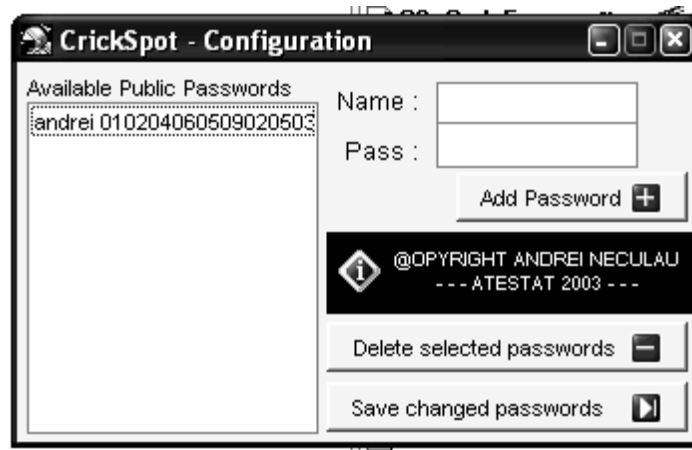
CrickSpot - Aplicatie Software  
pentru Criptare /P4  
Atestat 2003 - Andrei Neculau  
Colegiul National Vasile  
Alecsandri, Galati

## ÎN DETALIU

---

Aplicația software este conceput ca trei module ce acționează pe cont propriu: de configurare, pentru criptare și unul pentru decriptare. Toate trei prezintă o interfață ușor de accesat și conferă o utilitate maximă. Pentru a salva opțiuni și pentru a face un simplu transfer de date între module, se folosește un fișier INI, în care se scriu parolele publice, pozițiile ferestrelor, și alte opțiuni ale modulelor.

### Modulul de configurare



**Available Public Passwords** - parole publice deja introduse

**Name:** - introduceți un nume de cod pentru o nouă parolă

**Pass:** - introduceți noua parolă publică

**Add Password** - butonul care adaugă noua parolă

**Delete selected passwords** - butonul care șterge din listă parolele publice selectate

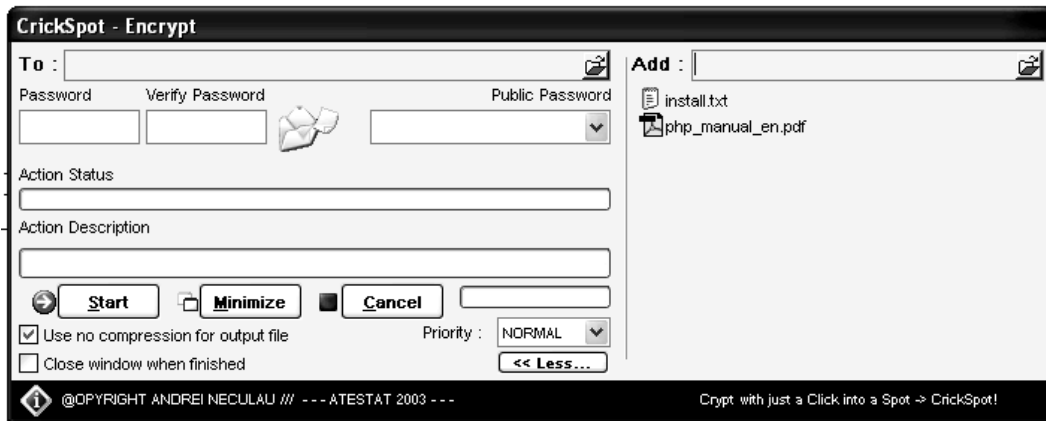
CrickSpot - Aplicatie Software  
pentru Criptare /Ps  
Atestat 2003 - Andrei Neculau  
Colegiul National Vasile  
Alecsandri, Galati

Save changed passwords – în cazul în care s-au făcut  
modificări, butonul salvează noile parole publice

---

CrickSpot - Aplicatie Software  
pentru Criptare /P6  
Atestat 2003 - Andrei Neculau  
Colegiul National Vasile  
Alecsandri, Galati

## Modulul pentru criptare



**To:** - se alege fișierul de destinație, ce va avea extensia .csf (CrickSpot file)

**Password** - introduceți parola pe care va trebui să o comunicați destinatarului

**Verify Password** - introduceți din nou aceeași parolă ; aceasta se cere din nou pentru siguranță

**Public Password** - selectați o parolă publică pe care ați ales-o în prealabil cu destinatarul, și pe care ați introdus-o cu ajutorul modului de configurare CrickSpot

**Action Status** - vă informează asupra stării actuale

**Action Description** - vă informează asupra acțiunii în curs

**Start/Pause** - butonul care activează, oprește temporar procesul de criptare

**Minimize** - butonul care duce aplicația ca iconiță în taskbar

**Close/Cancel** - butonul care închide definitiv aplicația

**Use no compression...** – opțiune de a folosi sau nu compresia

**Bzip** după criptarea fișierelor

CrickSpot - Aplicatie Software  
pentru Criptare /P7  
Atestat 2003 - Andrei Neculau  
Colegiul National Vasile  
Alecsandri, Galati

Close window when finished - în cazul în care aplicația este folosită de mai multe ori, pentru a ușura procesul, aplicația se poate închide automat când procesul de criptare se termină

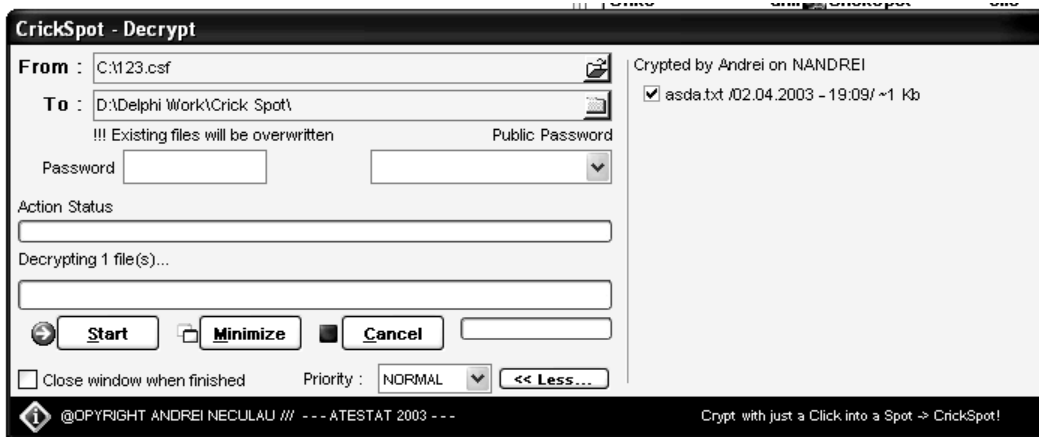
Priority - opțiune de a selecta prioritatea procesului

More/Less – buton pentru a afișa porțiunea de adăugare a fișierelor pentru criptare (optimizat pentru utilizare integrată în ®Microsoft Windows)

Add: se aleg fișierele care urmează a fi criptate

---

### Modulul pentru decriptare



From: - se alege fișierul cu extensia .csf (CrickSpot file)

To: - se alege directorul destinație, unde se vor decripta fișierele selectate (cele existente vor fi suprascrise)

Password - introduceți parola pe care v-a comunicat-o expeditorul fișierului CrickSpot

Public Password - selectați o parolă publică pe care ați ales-o în prealabil cu expeditorul, și pe care ați introdus-o cu ajutorul modului de configurare CrickSpot

CrickSpot - Aplicatie Software  
pentru Criptare /P8  
Atestat 2003 - Andrei Neculau  
Colegiul National Vasile  
Alecsandri, Galati



Action Status - vă informează asupra stării actuale

Action Description - vă informează asupra acțiunii în curs

Start/Pause - butonul care activează, ]oprește temporar procesul de decriptare

Minimize - butonul care duce aplicația ca iconiță în taskbar

Close/Cancel - butonul care închide definitiv aplicația

Close window when finished - în cazul în care aplicația este folosită de mai multe ori, pentru a ușura procesul, aplicația se poate închide automat cand procesul de decriptare se termină

Priority - opțiune de a selecta prioritatea procesului

More/Less - buton pentru a afișa porțiunea de adăugare a fișierelor pentru criptare (optimizat pentru utilizare integrată în ®Microsoft Windows)

Crypted by... - pentru o notă de autenticitate, fișierul CrickSpot conține și numele utilizatorului Windows, cât și numele computerului pe care s-a făcut criptarea

CrickSpot - Aplicatie Software  
pentru Criptare /P9  
Atestat 2003 - Andrei Neculau  
Colegiul National Vasile  
Alecsandri, Galati

## CODUL – Principalele funcții ale proceselor de criptare-decriptare

---

### Procedura de criptare (interfață și operații fizice)

#### **## interfață**

```
procedure TCodingForm.MainThreadExecute(Sender: TObject);
var sr: TSearchRec;
    destFile,dest2File: TFileStream;
    tempFile,sourceFile,dest: TMemoryStream;
    iCnt: LongInt;
    fileList: TCSFileList;
    sTemp: String;
    tempBuffer: PChar;
begin
```

#### **## se construiește un model (tabelă) după parola publică**

```
PublicPassToModel(PublicPass.Text);
```

#### **## inițializări, creez stream-uri de memorie pentru fișiere**

```
Timer.Enabled:=True;
fileList.NoFiles:=0;
sourceFile:=TMemoryStream.Create;
tempFile:=TMemoryStream.Create;
```

#### **## creez o listă cu fișierele ce sunt pregătite pentru criptare (anterior copiate în directorul temporar CrickSpot)**

```
ActionStatus.Caption:='Creating filelist...';
if FindFirst(CSDirectory+'\\*', faAnyFile, sr) = 0 then begin
repeat
if (sr.Name<>'') and (sr.Name<>'..') then
with fileList do begin
inc(NoFiles);
Names[NoFiles]:=sr.Name;
Sizes.Init[NoFiles]:=sr.Size;
end;
until FindNext(sr) <> 0;
FindClose(sr);
end;
```

#### **## inițializări, creez stream pentru fișierul destinație, pregătesc bara de progres, etc.**

CrickSpot - Aplicatie Software  
pentru Criptare /P10  
Atestat 2003 - Andrei Neculau  
Colegiul National Vasile  
Alecsandri, Galati

```
dest:=TMemoryStream.Create;  
ActionDescription.Caption:='Encrypting '+IntToStr(fileList.NoFiles)+' file(s)...';  
TotalProgress.Max:=fileList.NoFiles*100;
```

**## pentru fiecare fișier pregătit pentru criptare**

```
for iCnt:=1 to fileList.NoFiles do begin
```

**## actualizez bara de progres**

```
TotalProgressValue:=(iCnt-1)*100;  
ActionStatus.Caption:='Adding file: '"+fileList.Names[iCnt]+'";
```

**## încarc fișierul în memorie**

```
sourceFile.LoadFromFile(CSDirectory+'\'+fileList.Names[iCnt]);
```

**## atașez un număr de control pentru fișierul curent, ce va fi folosit pentru securitate la decriptare**

```
fileList.Sizes.Check[iCnt]:=GetMemoryChecksum(sourceFile);
```

**## încep criptarea fișierului din memorie**

```
CSCodeStream(sourceFile,tempFile,PassEdit.Text);
```

**## rețin mărimea rezultatului și salvez în memorie**

```
fileList.Sizes.Code[iCnt]:=tempFile.Size;  
tempFile.SaveToStream(dest);  
end;
```

**## adaug antete pentru arhiva de fișiere criptate**

```
ActionStatus.Caption:='Adding file headers...';  
sTemp:=Header1+Header2+SysInfo.CurrentUser+' on  
'+SysInfo.Computername+#250+Header3;  
sTemp:=sTemp+MakeFileHeader(fileList);  
sTemp:=sTemp+'|'+Header4+Header5;  
tempBuffer:=PChar(sTemp);
```

**## creez fișierul destinație și scriu din memorie**

```
destFile:=TFileStream.Create(CSDirectory+'\'+ExtractFilename(DestinationFile.FileName)+'.tmp',fmCreate);  
destFile.WriteBuffer(tempBuffer^,length(tempBuffer));  
dest.SaveToStream(destFile);
```

CrickSpot - Aplicatie Software  
pentru Criptare /P11  
Atestat 2003 - Andrei Neculau  
Colegiul National Vasile  
Alecsandri, Galati

**## actualizez interfața (procesul s-a terminat, deci butoanele se dezactivează)**

```
DoBtn.Enabled:=False;  
MinimizeBtn.Enabled:=False;  
ShrinkCheck.Enabled:=False;  
CloseCheck.Enabled:=False;  
PriorityBox.Enabled:=False;  
PriorityLabel.Enabled:=False;
```

**## în cazul în care a fost selectată opțiunea de a folosi compresie...**

```
if not ShrinkCheck.Checked then begin  
  ActionStatus.Caption:='Shrinking file...';
```

**## creez stream-ul pentru fișierul de destinație**

```
dest2File:=TFileStream.Create(DestinationFile.FileName,fmCreate);  
ShrinkProgress.Max:=destFile.Size;  
CancelBtn.Enabled:=False;
```

**## se face compresia**

```
DoCompress(destFile,dest2File);  
CancelBtn.Enabled:=True;  
dest2File.Free;  
ActionStatus.Caption:='Deleting temporary files...';
```

**## șterg arhiva temporară de fișiere compresate**

```
DeleteFile(CSDirectory+'\'+ExtractFilename(DestinationFile.FileName));  
End
```

**## ...altfel, dacă nu a fost selectată compresia, atunci copii arhiva temporară în fișierul de destinație**

```
else begin  
  destFile.Free;  
  CopyFile(PChar(CSDirectory+'\'+ExtractFilename(DestinationFile.FileName)+'.tmp'),  
PChar(DestinationFile.FileName),False);  
end;
```

---

**## operații fizice**

```
Function CSCodeStream(Source: TMemoryStream; var Dest: TMemoryStream;  
mdPassword: TModelPassword): Boolean;  
var bAllOK,bLastBlock: Boolean;
```

```
    CrickSpot - Aplicatie Software  
                pentru Criptare /P12  
    Atestat 2003 - Andrei Neculau  
    Colegiul National Vasile  
                Alecsandri, Galati
```

```
ReadBuff,WriteBuff: TByteBuffer;  
iReadBuffCnt,iWriteBuffCnt,iNumberOfReadBlocks,  
iCnt2,iCnt3,iCnt4,iSize: LongInt;  
hxStream,hxStream2,hxStream3: TDbIHexa;  
sTempStr,sTempStr2,sPassPattern: String;
```

**## funcția care citește din memorie blocuri de câte 10000 de caractere**

```
Function GetNextBlock:Boolean;  
begin  
  bLastBlock:=False;  
  FreeBuffer(ReadBuff);  
  if iSize-Source.Position<=0 then begin GetNextBlock:=False; Exit; end;  
  iReadBuffCnt:=Min(10000,iSize-Source.Position);  
  Source.ReadBuffer(ReadBuff,iReadBuffCnt);  
  GetNextBlock:=True;  
  if Source.Position=iSize then bLastBlock:=True;  
  inc(iNumberOfReadBlocks);  
end;
```

**## funcția care scrie în memorie blocuri de câte 10000 de caractere**

```
Procedure DumpBuffer(Forced:Boolean);  
var iCnt: Integer;  
begin  
  if (not Forced) and (iWriteBuffCnt>=10000) then begin  
    Dest.WriteBuffer(WriteBuff,10000);  
    for iCnt:=10001 to iWriteBuffCnt do  
      WriteBuff[iCnt-10000]:=WriteBuff[iCnt];  
    iWriteBuffCnt:=iWriteBuffCnt-10000;  
  end;  
  if Forced then Dest.WriteBuffer(WriteBuff,iWriteBuffCnt);  
end;  
begin  
  ProgressValue:=0;  
  iNumberOfReadBlocks:=-1;  
  bAllOK:=True;
```

**## se construiește modelul (tabela) după parola specifică arhivei curente**

```
MakeModelPassword(sPassPattern,mdPassword);  
try  
  try  
    sTempStr:=""; iWriteBuffCnt:=0;
```

CrickSpot - Aplicatie Software  
pentru Criptare /P13  
Atestat 2003 - Andrei Neculau  
Colegiul National Vasile  
Alecsandri, Galati

```
Dest.Clear;  
iSize:=Source.Size;  
Source.Position:=0;
```

```
## cât timp voi putea citi blocuri de caractere (nu am ajuns la sfârșitul stream-ului)
```

```
while GetNextBlock do begin
```

```
## pentru fiecare caracter...
```

```
for iCnt2:=1 to iReadBuffCnt do  
begin
```

```
## actualizez bara de progres
```

```
ProgressValue:=(iNumberOfReadBlocks*10000 + iCnt2)*100/iSize;
```

```
## transform codul ASCII din baza 10 în bază 16
```

```
hxStream:=DecimalToHexadecimal(ReadBuff[iCnt2]);
```

```
## urmează o transformare a literelor bazei 16 în cifre
```

```
if hxStream[1] in ['A'..'F'] then hxStream2:=IntToStr(ord(hxStream[1])-55)  
else hxStream2:='0'+hxStream[1];
```

```
hxStream3:=hxStream2;
```

```
if hxStream[2] in ['A'..'F'] then hxStream2:=IntToStr(ord(hxStream[2])-55)  
else hxStream2:='0'+hxStream[2];
```

```
hxStream3:=hxStream3+hxStream2;
```

```
## atașez rezultatul „hexa” unui șir temporar prin transformarea prin părți cu ajutorul funcției MultipleThree (explicată mai jos)
```

```
sTempStr:=sTempStr+MultipleThree(StrToInt(hxStream3[1]+hxStream3[2]))+  
MultipleThree(StrToInt(hxStream3[3]+hxStream3[4]));
```

```
## dacă am ajuns la un bloc de lungime 24, sau nu mai sunt caractere de citit...
```

```
if (length(sTempStr)=24) or ((iCnt2=iReadBuffCnt) and (bLastBlock)) then begin
```

```
## aplic modelul parolei asupra acestui bloc
```

```
sTempStr:=ApplyModel(sPassPattern,sTempStr);
```

```
## încep scrierea rezultatului
```

```
iWriteBuffCnt:=iWriteBuffCnt+1;
```

```
WriteBuff[iWriteBuffCnt]:=length(sTempStr);
```

```
CrickSpot - Aplicatie Software  
pentru Criptare /P14  
Atestat 2003 - Andrei Neculau  
Colegiul National Vasile  
Alecsandri, Galati
```

```

    for iCnt3:=1 to length(sTempStr) do
    begin

## transform din baza 10 în baza 2 numărul de ordine al elementelor
        sTempStr2:=TenToBase(2,ord(sTempStr[iCnt3]));

## înlocuiesc secvențe de 0 și 1 cu ajutorul funcției ReplaceOneAndZero (explicată mai jos) pentru a reduce lungimea blocului
        sTempStr2:=ReplaceOneAndZero(sTempStr2);
        sTempStr2:=sTempStr2+#250;
        for iCnt4:=1 to length(sTempStr2) do begin
            iWriteBuffCnt:=iWriteBuffCnt+1;
            WriteBuff[iWriteBuffCnt]:=ord(sTempStr2[iCnt4]);
        end;
    end;

## încerc scrierea în memorie, atunci când este cazul
        DumpBuffer((bLastBlock) and (iCnt2=iReadBuffCnt));
        sTempStr:="";
    end;
end;
except
    bAllOK:=false;
end;
finally
    Result:=bAllOK;
end;
end;

```

---

## Procedura de decriptare (interfață și operații fizice)

### ## interfață

```

procedure TDecodingForm.MainThreadExecute(Sender: TObject);
var tempFile,destFile: TFileStream;
    sourceStream,destStream,dest: TMemoryStream;
    buff: TByteBuffer;
    sTemp: String;
    iCnt,iSize,iNumberOfFiles: Integer;
    fileList: TCSFileList;

```

CrickSpot - Aplicatie Software  
 pentru Criptare /P15  
 Atestat 2003 - Andrei Neculau  
 Colegiul National Vasile  
 Alecsandri, Galati

```
begin
```

```
## se construiește un model (tabelă) după parola publică
```

```
PublicPassToModel(PublicPass.Text);
```

```
## stabilesc un bloc de maxim 3000 de caractere unde trebuie să găsesc tot antetul
```

```
iSize:=Min(3000,FileSizeByName(SourceFile.Filename));
```

```
## dacă fișierul nu e nici comprimat cu Bzip și nici nu e un fișier CrickSpot...
```

```
if (not IsBzip(SourceFile.Filename)) and (not IsPlain(SourceFile.Filename)) then begin
```

```
  ActionStatus.Caption:='File is not a CrickSpot one !';
```

```
  CancelBtn.Caption:='&Close';
```

```
  DoBtn.Enabled:=False;
```

```
  MinimizeBtn.Enabled:=False;
```

```
  CloseCheck.Enabled:=False;
```

```
  PriorityBox.Enabled:=False;
```

```
End
```

```
## ...altfel...
```

```
else begin
```

```
## dacă fișierul e comprimat
```

```
if IsBzip(SourceFile.Filename) then begin
```

```
## creez stream-uri de memorie pentru sursă și destinație
```

```
tempFile:=TFileStream.Create(SourceFile.Filename,fmOpenRead);
```

```
destFile:=TFileStream.Create(CSDirectory+'\' +ExtractFilename(SourceFile.Filename) +'.tmp',fmCreate);
```

```
## dezarhivez fișierul comprimat
```

```
DoUncompress(tempFile,destFile);
```

```
## golesc memoria
```

```
FreeAndNil(destFile);
```

```
FreeAndNil(tempFile);
```

CrickSpot - Aplicatie Software  
pentru Criptare /P16  
Atestat 2003 - Andrei Neculau  
Colegiul National Vasile  
Alecsandri, Galati



**## dacă fișierul dezarhivat nu este un fișier CrickSpot**

```
if not IsPlain(CSDirectory+'\' + ExtractFilename(SourceFile.Filename)+'.tmp') then
begin
  ActionStatus.Caption:='File is not a CrickSpot one !';
  CancelBtn.Caption:='&Close';
  DoBtn.Enabled:=False;
  MinimizeBtn.Enabled:=False;
  CloseCheck.Enabled:=False;
  PriorityBox.Enabled:=False;
  PriorityLabel.Enabled:=False;
  DeleteFile(CSDirectory+'\' + ExtractFilename(SourceFile.Filename)+'.tmp');
  exit;
end;
dest:=TMemoryStream.Create;
dest.LoadFromFile(CSDirectory+'\' + ExtractFilename(SourceFile.Filename)+'.tmp');
end
```

**## ...altfel, încarc în memorie fișierul CrickSpot**

```
else begin
  dest:=TMemoryStream.Create;
  dest.LoadFromFile(SourceFile.Filename);
end;
Timer.Enabled:=True;
```

**## încerc citirea antetului din fișier**

```
dest.ReadBuffer(buff,iSize);
sTemp:="";
for iCnt:=iSize downto 1 do
  if buff[iCnt]=ord('|') then break;
for iSize:=1 to iCnt do sTemp:=sTemp+chr(buff[iSize]);
```

**## încerc să extrag informațiile din antet**

```
CSReadHeader(sTemp,fileList);
```

**## marchează sfârșitul antetului în fișierul din memorie**

```
dest.Seek(iCnt+length(Header4)+length(Header5),soFromBeginning);
```

**## actualizează bara de progres**

```
TotalProgress.Max:=fileList.NoFiles*100;
```

CrickSpot - Aplicatie Software  
pentru Criptare /P17  
Atestat 2003 - Andrei Neculau  
Colegiul National Vasile  
Alecsandri, Galati

**## în cazul în care nu au fost selectate fișiere în panoul de selecție atunci decriptez toate fișierele conținute în arhiva CrickSpot**

```
if SelFiles.Items.Count=0 then iNumberOfFiles:=fileList.NoFiles
```

**## ...altfel memorez fișierele pregătite pentru decriptare**

```
else begin
```

```
  iNumberOfFiles:=0;
```

```
  for iCnt:=0 to SelFiles.Items.Count-1 do
```

```
    if SelFiles.Checked[iCnt] then inc(iNumberOfFiles);
```

```
  end;
```

```
  ActionDescription.Caption:='Decrypting '+IntToStr(iNumberOfFiles)+' file(s)...!';
```

**## pentru fiecare fișier pregătit pentru decriptare**

```
for iCnt:=1 to fileList.NoFiles do
```

**## dacă a fost pregătit sau nu a fost selectat niciunul (fișier CrickSpot comprimat)**

```
if ((SelFiles.Items.Count>0) and (SelFiles.Checked[iCnt-1])) or  
(SelFiles.Items.Count=0) then
```

```
  begin
```

**## actualizez bara de progres, inițializări, creez stream-uri de memorie**

```
  TotalProgressValue:=(iCnt-1)*100;
```

```
  ActionStatus.Caption:='Extracting file: '"+fileList.Names[iCnt]+'!';
```

```
  sourceStream:=TMemoryStream.Create;
```

```
  destStream:=TMemoryStream.Create;
```

**## copii din stream-ul sursă doar acea parte care este asociată fișierului curent**

```
  sourceStream.CopyFrom(dest,fileList.Sizes.Code[iCnt]);
```

```
  sourceStream.Seek(0,soFromBeginning);
```

**## încep decriptarea fișierului din memorie; în cazul în care nu e un succes...**

```
if not CSDecodeStream(sourceStream,destStream,PassEdit.Text) then begin
```

```
  FreeAndNil(dest);
```

```
  DoBtn.Enabled:=False;
```

```
  MinimizeBtn.Enabled:=False;
```

```
  CloseCheck.Enabled:=False;
```

```
  PriorityBox.Enabled:=False;
```

CrickSpot - Aplicatie Software

pentru Criptare /P18

Atestat 2003 - Andrei Neculau

Colegiul National Vasile

Alecsandri, Galati

```
PriorityLabel.Enabled:=False;  
CancelBtn.Caption:='&Close';
```

```
ActionStatus.Caption:='Bad password or damaged file!';  
Exit;  
end;
```

**## în cazul în care decriptarea s-a terminat cu bine, fac testul de siguranță prin numărul de control al caracterelor din fișier și...**

```
if GetMemoryChecksum(destStream)=fileList.Sizes.Check[iCnt] then
```

**## ...salvez stream-ul de memorie într-un fișier**

```
destStream.SaveToFile(DestinationDir.Path+fileList.Names[iCnt])  
else begin
```

**## ...altfel, golesc memoria și dezactivez interfața**

```
FreeAndNil(dest);  
DoBtn.Enabled:=False;  
MinimizeBtn.Enabled:=False;  
CloseCheck.Enabled:=False;  
PriorityBox.Enabled:=False;  
PriorityLabel.Enabled:=False;  
CancelBtn.Caption:='&Close';
```

```
ActionStatus.Caption:='Bad password or damaged file!';  
Exit;  
end;  
FreeAndNil(destStream);  
FreeAndNil(sourceStream);  
end;
```

**## actualizez interfața (procesul s-a terminat, deci butoanele se dezactivează)**

```
FreeAndNil(dest);  
DoBtn.Enabled:=False;  
MinimizeBtn.Enabled:=False;  
CloseCheck.Enabled:=False;  
PriorityBox.Enabled:=False;  
PriorityLabel.Enabled:=False;
```

```
ActionStatus.Caption:='Done!';
```

CrickSpot - Aplicatie Software  
pentru Criptare /P19  
Atestat 2003 - Andrei Neculau  
Colegiul National Vasile  
Alecsandri, Galati

```
end;  
end;
```

---

### **## operații fizice**

```
Function CSDecodeStream(Source: TMemoryStream; var Dest: TMemoryStream;  
mdPassword: TModelPassword): Boolean;  
var bAllOK,bFinished,bModelFinished,bLastBlock,bTemp: Boolean;  
    ReadBuff,WriteBuff: TByteBuffer;  
    iReadBuffCnt,iWriteBuffCnt,iModelCnt,iNumberOfReadBlocks,  
    iCnt2,iCnt3,iCnt4,iCnt5,iTempCnt,iSize: LongInt;  
    sTempStr,sTempStr2,sCodeStr,sPassPattern: String;
```

### **## funcția care citește din memorie blocuri de câte 10000 de caractere**

```
Function GetNextBlock:Boolean;  
begin  
    FreeBuffer(ReadBuff);  
    if bLastBlock then begin GetNextBlock:=False; Exit; end;  
    bLastBlock:=False;  
    iReadBuffCnt:=Min(10000,iSize-Source.Position);  
    Source.ReadBuffer(ReadBuff,iReadBuffCnt);  
    GetNextBlock:=True;  
    inc(iNumberOfReadBlocks);  
    if Source.Position=Source.Size then bLastBlock:=True;  
end;
```

### **## funcția care scrie în memorie blocuri de câte 10000 de caractere**

```
Procedure DumpBuffer(Forced:Boolean);  
var iCnt: Integer;  
begin  
    if (not Forced) and (iWriteBuffCnt>=10000) then begin  
        Dest.WriteBuffer(WriteBuff,10000);  
        for iCnt:=10001 to iWriteBuffCnt do  
            WriteBuff[iCnt-10000]:=WriteBuff[iCnt];  
        iWriteBuffCnt:=iWriteBuffCnt-10000;  
    end;  
    if Forced then Dest.WriteBuffer(WriteBuff,iWriteBuffCnt);  
end;  
begin  
    ProgressValue:=0;  
    iNumberOfReadBlocks:=-1; bLastBlock:=False;
```

```
    CrickSpot - Aplicatie Software  
                pentru Criptare /P20  
    Atestat 2003 - Andrei Neculau  
                Colegiul National Vasile  
                Alecsandri, Galati
```

**## se construiește modelul (tabela) după parola specifică arhivei curente**

```
MakeModelPassword(sPassPattern,mdPassword);
bAllOK:=true;
try
  try
    iTempCnt:=0; sTempStr:=""; iWriteBuffCnt:=0; iReadBuffCnt:=0; iModelCnt:=0;
    bFinished:=true; sCodeStr:=""; bModelFinished:=true; bTemp:=False;
    Dest.Clear;
    Dest.Position:=0;
    iSize:=Source.Size;
```

**## cât timp voi putea citi blocuri de caractere (nu am ajuns la sfârșitul stream-ului)**

```
while GetNextBlock do begin
  iCnt2:=0;
```

**## controlez dacă sunt la începutul creării blocului temporar 1**

```
if (bFinished) and (bModelFinished) and (not bTemp) then begin
iModelCnt:=ReadBuff[1]; inc(iCnt2); end;
  bTemp:=false;
  repeat
    bModelFinished:=false;
    repeat
      bFinished:=false;
    repeat
```

**## actualizez bara de progres**

```
  ProgressValue:=(iNumberOfReadBlocks*10000 + iReadBuffCnt)*100/iSize;
  inc(iCnt2);
```

**## citesc până nu mai am caractere sau am completat blocul temporar 2**

```
  if ReadBuff[iCnt2]<>250 then sCodeStr:=sCodeStr+chr(ReadBuff[iCnt2])
    else bFinished:=true;
  until (ReadBuff[iCnt2]=250) or (iCnt2=iReadBuffCnt);
  bModelFinished:=false;
```

**## dacă am completat blocul temporar 2**

```
  if (bFinished) then begin
```

CrickSpot - Aplicatie Software  
pentru Criptare /P21  
Atestat 2003 - Andrei Neculau  
Colegiul National Vasile  
Alecsandri, Galati

**## refac pașii inverși din procesul de criptare: refac secvențele de 0 și 1, aduc numerele din baza 2 în baza 10**

```
sTempStr2:=sTempStr2+chr(BaseToTen(2,RestoreOneAndZero(sCodeStr)));
sCodeStr:="";
inc(iTempCnt);
end;
if iTempCnt=iModelCnt then bModelFinished:=true;
until (bModelFinished) or (iCnt2=iReadBuffCnt);
```

**## dacă am completat blocul temporar 1**

```
if bModelFinished then begin
```

**## refac blocul inițial prin eliminarea modelului parolei de deasupra blocului**

```
sTempStr2:=RemoveModel(sPassPattern,sTempStr2);
for iCnt3:=1 to length(sTempStr2) do
```

**## dacă am un bloc de lungime 6, adică exact ¼ din mărimea blocului de lungime 24, din procesul de criptare...**

```
if iCnt3 mod 6=1 then begin
```

**## refac componentele inițiale prin aplicarea funcției inverse**

**RestoreMultipleThree**

```
iCnt4:=RestoreMultipleThree(sTempStr2[iCnt3]+sTempStr2[iCnt3+1]+sTempStr2[iCnt3+2]);
```

```
iCnt5:=RestoreMultipleThree(sTempStr2[iCnt3+3]+sTempStr2[iCnt3+4]+sTempStr2[iCnt3+5]);
```

**## urmează transformarea din baza 16 în baza 10**

```
iCnt4:=HexadecimalToDecimal_Basic(iCnt4,iCnt5);
```

**## încep scrierea rezultatului**

```
iWriteBuffCnt:=iWriteBuffCnt+1;
```

```
WriteBuff[iWriteBuffCnt]:=iCnt4;
```

**## încerc scrierea în memorie, atunci când este cazul**

```
DumpBuffer((iCnt2=iReadBuffCnt) and (bLastBlock) and (iCnt3+6>length(sTempStr2)))
```

```
end;
```

```
sTempStr2:="";
```

CrickSpot - Aplicatie Software  
pentru Criptare /P22  
Atestat 2003 - Andrei Neculau  
Colegiul National Vasile  
Alecsandri, Galati

```

        iTempCnt:=0;
        inc(iCnt2);
        bTemp:=False;
        if iCnt2<=iReadBuffCnt then begin iModelCnt:=ReadBuff[iCnt2]; bTemp:=True;
end;
        end;
        until iCnt2>=iReadBuffCnt;
        end;
    except
        bAlLOK:=false;
    end;
finally
    Result:=bAlLOK;
end;
end;

```

---

## Proceduri și funcții secundare, ce ajută procesele de (de)criptare

### ## funcția transformă din baza 10 în baza 16

```

Function DecimalToHexadecimal(wNumber: Word): TDbIHexa;
const HexChars: Array[0..$F] of Char='0123456789ABCDEF';
var hxResultHexa: TDbIHexa;
begin
    hxResultHexa:=HexChars[Hi(wNumber) shr 4]+HexChars[Hi(wNumber) and $F]+
        HexChars[Lo(wNumber) shr 4]+HexChars[Lo(wNumber) and $F];
    repeat
        if hxResultHexa[1]='0' then Delete(hxResultHexa,1,1);
    until (hxResultHexa[1]<>'0') or (length(hxResultHexa)=2);
    DecimalToHexadecimal:=UpperCase(hxResultHexa);
end;

```

### ## funcția transformă din baza 16 în baza 10

```

Function HexadecimalToDecimal_Basic(First,Second:Byte): Byte;
begin
    HexadecimalToDecimal_Basic:=Second+First*16;
end;

```

### ## funcția returnează procentajul

```

Function Procent(iPart,iTotal: LongInt): TOneHundred;
begin
    Procent:=(iPart * 100) div iTotal;

```

CrickSpot - Aplicatie Software  
 pentru Criptare /P23  
 Atestat 2003 - Andrei Neculau  
 Colegiul National Vasile  
 Alecsandri, Galati

```
end;
```

### **## funcția transformă din baza 10 în orice bază <10**

```
Function TenToBase(btBase: TBase; iNumber: Integer): String;  
var iCnt: Integer;  
    sResult: String;  
begin  
    sResult:="";  
    repeat  
        iCnt:=iNumber div btBase;  
        sResult:=IntToStr(iNumber mod btBase)+sResult;  
        iNumber:=iCnt;  
    until iNumber<btBase;  
    if iNumber>0 then sResult:=IntToStr(iNumber)+sResult;  
    TenToBase:=sResult;  
end;
```

### **## funcția transformă din orice bază <10 în baza 10**

```
Function BaseToTen(btBase: TBase; sNumber: String): Integer;  
var iCnt,iResult: Integer;  
    sTempStr: String;  
begin  
    iResult:=StrToInt(sNumber[length(sNumber)]);  
    sTempStr:="";  
    for iCnt:=length(sNumber) downto 1 do sTempStr:=sTempStr+sNumber[iCnt];  
    for iCnt:=2 to length(sTempStr) do  
        iResult:=iResult+(Round(Power(btBase,iCnt-1))*StrToInt(sTempStr[iCnt]));  
    BaseToTen:=iResult;  
end;
```

### **## face legătura dintre un fișier și un stream de memorie, căruia i se calculează numărul de control**

```
Function GetFileChecksum(sFilename: String): LongInt;  
var iResult: LongInt;  
    fInput: TMemoryStream;  
begin  
    try  
        iResult:=0;  
        fInput:=TMemoryStream.Create;  
        fInput.LoadFromFile(sFilename);  
        iResult:=GetMemoryChecksum(fInput);
```

CrickSpot - Aplicatie Software  
pentru Criptare /P24  
Atestat 2003 - Andrei Neculau  
Colegiul National Vasile  
Alecsandri, Galati



```
    GetFileChecksum:=iResult;  
except;  
    GetFileChecksum:=FileError;  
end;  
end;
```

### **## funcția care calculează numărul de control pentru un stream de memorie**

```
Function GetMemoryChecksum(Src: TMemoryStream): LongInt;  
var iResult,iSize,iCnt1: LongInt;  
    fInput: TMemoryStream;  
    chTemp: Char;  
begin  
    fInput:=TMemoryStream.Create;  
    fInput.Clear;  
    fInput.CopyFrom(Src,0);  
    fInput.Position:=0;  
    iResult:=0;  
    iSize:=fInput.Size;  
    for iCnt1:=1 to iSize do  
    begin  
        fInput.ReadBuffer(chTemp,1);  
        iResult:=iResult+ord(chTemp);  
    end;  
    fInput.Free;  
    GetMemoryChecksum:=iResult;  
end;
```

### **## mărimea unui fișier**

```
Function FileSizeByName(sFilename: String):LongInt;  
var lngiResult: LongInt;  
    fInput: TMemoryStream;  
begin  
    try  
        fInput:=TMemoryStream.Create;  
        fInput.LoadFromFile(sFilename);  
        lngiResult:=fInput.Size;  
        FileSizeByName:=lngiResult;  
    except  
        FileSizeByName:=FileError;  
    end;
```

CrickSpot - Aplicatie Software  
pentru Criptare /P25  
Atestat 2003 - Andrei Neculau  
Colegiul National Vasile  
Alecsandri, Galati

```
end;
```

### **## funcția care controlează validitatea unui fișier**

```
Function IsPathValid(sPath: String): Boolean;  
var iCnt: Integer;  
begin  
  for iCnt:=1 to length(sPath) do  
    if sPath[iCnt] in [':','*','?','"', '<', '>', '|'] then  
      if (sPath[iCnt]<>':') and (sPath[iCnt+1]<>'\\') and (iCnt<>2) then  
        begin IsPathValid:=false; exit; end;  
      IsPathValid:=true;  
    end;  
  end;
```

### **## funcția care golește memoria tampon**

```
Procedure FreeBuffer(var Buff: TByteBuffer);  
var iCnt: Integer;  
begin  
  for iCnt:=1 to 10001 do  
    Buff[iCnt]:=0;  
  end;
```

### **## funcția care construiește un model (tabelă) după o parolă dată, în funcție de modelul parolei publice**

```
Procedure MakeModelPassword(var sPassPattern: String; mdPassword:  
TModelPassword);  
var iCnt1,iCnt2,iCnt3,iCnt4,iCnt5: Integer;  
begin  
  if length(mdPassword)=0 then mdPassword:='CrickSpot';  
  for iCnt1:=1 to 4 do  
    for iCnt2:=1 to 6 do  
      if Model[iCnt1,iCnt2]>length(mdPassword)  
        then begin  
          TempModel[iCnt1,iCnt2]:=Model[iCnt1,iCnt2];  
          TempModel[iCnt1,iCnt2]:=TempModel[iCnt1,iCnt2] mod length(mdPassword);  
        end  
        else TempModel[iCnt1,iCnt2]:=Model[iCnt1,iCnt2];  
  sPassPattern:="";  
  for iCnt3:=1 to 4 do  
    for iCnt4:=1 to 6 do begin  
      iCnt5:=ord(mdPassword[TempModel[iCnt3,iCnt4]]);
```

CrickSpot - Aplicatie Software  
pentru Criptare /P26  
Atestat 2003 - Andrei Neculau  
Colegiul National Vasile  
Alecsandri, Galati

```
sPassPattern:=sPassPattern+StringOfChar('0',3-  
length(IntToStr(iCnt5)))+IntToStr(iCnt5);  
end;  
end;
```

### **## golește lista de fișiere**

```
Procedure FreeFileList(var FileList: TCSFileList);  
var iCnt: Integer;  
begin  
for iCnt:=1 to 100 do  
with FileList do  
begin  
Names[iCnt]:= "";  
Dates[iCnt]:= "";  
Times[iCnt]:= "";  
Sizes.Init[iCnt]:= 0;  
Sizes.Code[iCnt]:= 0;  
Sizes.Check[iCnt]:= 0;  
NoFiles:= 0;  
end;  
end;
```

### **## funcția care construiește modelul parolei publice**

```
Procedure PublicPassToModel(PP: String);  
var iCnt1,iCnt2,iTemp: Integer;  
begin  
Delete(PP,1,pos(' ',PP));  
iCnt1:=1; iCnt2:=1;  
while length(PP)>0 do begin  
iTemp:=StrToInt(PP[1]+PP[2]);  
Model[iCnt1,iCnt2]:=iTemp;  
Delete(PP,1,2);  
inc(iCnt2);  
if iCnt2=7 then begin inc(iCnt1); iCnt2:=1; end;  
end;  
end;
```

```
Function AddZ(sText: String): String;  
var sTemp,sResult: String;  
iCnt,iZ: Integer;  
begin
```

CrickSpot - Aplicatie Software  
pentru Criptare /P27  
Atestat 2003 - Andrei Neculau  
Colegiul National Vasile  
Alecsandri, Galati

```

iZ:=0;
sResult:="";
for iCnt:=1 to length(sText) do begin
  sTemp:="";
  if iZ>0 then sTemp:='z'+IntToStr(iZ);
  if iZ>8 then begin sResult:=sResult+sTemp; sTemp:=""; iZ:=0; end;
  if sText[iCnt]='0' then inc(iZ)
  else begin
    if iZ>0 then begin sResult:=sResult+sTemp; sTemp:=""; iZ:=0; end;
    sResult:=sResult+sText[iCnt];
  end;
end;
AddZ:=sResult;
end;

```

```

Function CodeOrd(sPattern,sText: String): String;
var iCnt,iNew: Integer;
    sResult: String;
begin
  sResult:="";
  for iCnt:=1 to Min(length(sPattern),length(sText)) do begin
    iNew:=StrToInt(sPattern[iCnt]) - StrToInt(sText[iCnt]);
    if iNew<=0 then sResult:=sResult+chr(100-iNew)
    else sResult:=sResult+IntToStr(iNew);
  end;
  CodeOrd:=sResult;
end;

```

```

Function CodeGroup(sText: String): String;
var sResult: String;
    iCnt,iStep,iNew: Integer;
begin
  iStep:=1;
  sResult:="";
  iCnt:=0;
  repeat
    inc(iCnt,iStep);
    iStep:=1;
    if not (sText[iCnt] in ['1'..'9']) then sResult:=sResult+sText[iCnt]
    else
      if iCnt<length(sText) then begin

```

CrickSpot - Aplicatie Software  
 pentru Criptare /P28  
 Atestat 2003 - Andrei Neculau  
 Colegiul National Vasile  
 Alecsandri, Galati

```

    iNew:=StrToIntDef(sText[iCnt]+sText[iCnt+1],-1);
    if iNew>-1 then begin sResult:=sResult+chr(iNew); iStep:=2; end
    else if StrToIntDef(sText[iCnt],-1)>-1 then
sResult:=sResult+chr(StrToInt(sText[iCnt]))
    end
    else if StrToIntDef(sText[iCnt],-1)>-1 then
sResult:=sResult+chr(StrToInt(sText[iCnt]))
    until iCnt+iStep>length(sText);
    CodeGroup:=sResult
end;

```

**## funcția care transformă un număr în formatul (număr=3xCÂT + REST)  
[REST (2^REST + CÂT) ]**

```

Function MultipleThree(byNumber: TOneHundred): TDbfHexa;
var byMultiple,byNo1,byNo2: Byte;
    hxResult: TDbfHexa;
begin
    byNo1:=byNumber mod 3;
    byMultiple:=byNumber div 3;
    byNo2:=Round(Power(2,byNo1))+byMultiple;
    hxResult:=IntToStr(byNo1)+IntToStr(byNo2);
    if byNo2<10 then Insert('0',hxResult,2);
    MultipleThree:=hxResult;
end;

```

**## funcția care suprapune modelul parolei asupra unui bloc de text**

```

Function ApplyModel(sPassPattern,sText: String): String;
var iCnt,iCnt2: Integer;
    sResult: String;
begin
    sResult:="";
    for iCnt:=1 to length(sText) do begin
        iCnt2:=ord(sText[iCnt]);
        sResult:=sResult+StringOfChar('0',3-length(IntToStr(iCnt2)))+IntToStr(iCnt2);
    end;
    sResult:=CodeOrd(sPassPattern,sResult);
    sResult:=AddZ(sResult);
    sResult:=CodeGroup(sResult);
    ApplyModel:=sResult;
end;

```

CrickSpot - Aplicatie Software  
 pentru Criptare /P29  
 Atestat 2003 - Andrei Neculau  
 Colegiul National Vasile  
 Alecsandri, Galati

**## funcția care înlocuiește secvențele de 0 și 1 prin lungimea secvenței**

```
function ReplaceOneAndZero(sNumberBaseTwo: String): String;
var iCnt1,iCnt2: Integer;
    sTempStr: String;
    chLast: Char;
begin
if sNumberBaseTwo<>" then begin
sNumberbaseTwo:=sNumberBaseTwo+' ';
iCnt2:=0;
chLast:=sNumberBaseTwo[1];
iCnt1:=0;
repeat
iCnt1:=iCnt1+1;
if (chLast=sNumberBaseTwo[iCnt1]) and (iCnt2+1<9)
then iCnt2:=iCnt2+1
else if chLast<>' ' then begin
if chLast='0' then iCnt2:=-iCnt2;
sTempStr:=sTempStr+chr(iCnt2+100);
iCnt2:=0;
chLast:=sNumberBaseTwo[iCnt1];
iCnt1:=iCnt1-1;
end;
until iCnt1=length(sNumberBaseTwo);
ReplaceOneAndZero:=sTempStr;
end
else ReplaceOneAndZero:="";
end;
```

**## funcția care construiește blocul pentru antet dintr-un text**

```
Function CSHeadString(sString: String): String;
var sResult: String;
    iCnt: Integer;
begin
sResult:="";
for iCnt:=1 to length(sString) do
sResult:=sResult+MultipleThree(ord(sString[iCnt])-45);
CSHeadString:=sResult;
end;
```

**## funcția care construiește antetul**

```
Function MakeFileHeader(fileList: TCSFileList): String;

CrickSpot - Aplicatie Software
           pentru Criptare /P30
Atestat 2003 - Andrei Neculau
           Colegiul National Vasile
           Alecsandri, Galati
```

```

var sResult2,sResult: String;
    w1,w2,w3: Word;
    b1,b2: Byte;
    Curr: Integer;
begin
for Curr:=1 to fileList.NoFiles do begin
with fileList do begin
{filedate}
    DecodeDate(Date,w1,w2,w3);
    b1:=length(IntToStr(w2))-2;
    b1:=b1*(-1);
    b2:=length(IntToStr(w3))-2;
    b2:=b2*(-1);
    Dates[Curr]:=StringOfChar('0',b2)+IntToStr(w3)+'.'+
StringOfChar('0',b1)+IntToStr(w2)+'.'+IntToStr(w1);
    Dates[Curr]:=CSHeadString(Dates[Curr]);
    DecodeTime(Time,w1,w2,w3,w3);
    b1:=length(IntToStr(w1))-2;
    b1:=b1*(-1);
    b2:=length(IntToStr(w2))-2;
    b2:=b2*(-1);
    Times[Curr]:=StringOfChar('0',b1)+IntToStr(w1)+'.'+
StringOfChar('0',b2)+IntToStr(w2);
    Times[Curr]:=CSHeadString(Times[Curr]);
{filename}
    Names[Curr]:=CSHeadString(Names[Curr]);
{final}
    sResult:='|'+Names[Curr]+'d'+Dates[Curr]+'t'+Times[Curr];
    sResult:=sResult+'i'+CSHeadString(IntToStr(Sizes.Init[Curr]))+
        'c'+CSHeadString(IntToStr(Sizes.Code[Curr]))+
        's'+CSHeadString(IntToStr(Sizes.Check[Curr]));
end;
sResult2:=sResult2+sResult;
end;
MakeFileHeader:=sResult2;
end;

```

CrickSpot - Aplicatie Software  
pentru Criptare /P31  
Atestat 2003 - Andrei Neculau  
Colegiul National Vasile  
Alecsandri, Galati

---

**Proceduri și funcții complementare specifice decriptării, ce refac structurile inițiale în funcție de rezultatele funcțiilor de mai sus**

```
Function CutZ(sText: String): String;
var iCnt: Integer;
    sResult: String;
begin
    sResult:="";
    iCnt:=1;
    repeat
        if sText[iCnt]='z' then begin
            sResult:=sResult+StringOfChar('0',StrToInt(sText[iCnt+1]));
            inc(iCnt,2);
        end
        else begin sResult:=sResult+sText[iCnt]; inc(iCnt); end;
    until iCnt>length(sText);
    CutZ:=sResult;
end;
```

```
Function DecodeGroupOrd(sPattern,sText: String): String;
var iCnt: Integer;
    sResult,sResult2: String;
begin
    sResult:="";
    for iCnt:=1 to length(sText) do
        if (ord(sText[iCnt])>99) then
            sResult:=sResult+sText[iCnt]
        else sResult:=sResult+IntToStr(ord(sText[iCnt]));

    sResult:=CutZ(sResult);
    sResult2:="";
    for iCnt:=1 to min(length(sPattern),length(sResult)) do begin
        if ord(sResult[iCnt])>99 then
            sResult2:=sResult2+IntToStr( StrToInt(sPattern[iCnt]) +ord(sResult[iCnt]) -100)
        else
            sResult2:=sResult2+IntToStr( StrToInt(sPattern[iCnt]) -StrToInt(sResult[iCnt]) )
    end;
    DecodeGroupOrd:=sResult2;
```

CrickSpot - Aplicatie Software  
pentru Criptare /P32  
Atestat 2003 - Andrei Neculau  
Colegiul National Vasile  
Alecsandri, Galati



```
end;
```

```
Function RestoreOneAndZero(sReplacedOneAndZero: String): String;  
var sResult: String;  
    iCnt1,iCnt2: Integer;  
begin  
    sResult:="";  
    for iCnt1:=1 to length(sReplacedOneAndZero) do  
        sResult:=sResult+IntToStr((ord(sReplacedOneAndZero[iCnt1])-100));  
        sReplacedOneAndZero:=sResult;  
        sResult:="";  
        iCnt1:=1;  
        repeat  
            if sReplacedOneAndZero[iCnt1]='-' then  
                begin  
                    sResult:=sResult+StringOfChar('0',StrToInt(sReplacedOneAndZero[iCnt1+1]));  
                    iCnt2:=2;  
                end  
            else  
                begin  
                    sResult:=sResult+StringOfChar('1',StrToInt(sReplacedOneAndZero[iCnt1]));  
                    iCnt2:=1;  
                end;  
            iCnt1:=iCnt1+iCnt2;  
        until iCnt1>length(sReplacedOneAndZero);  
        RestoreOneAndZero:=sResult;  
    end;
```

```
Function RemoveModel(sPassPattern,sText: String): String;  
var iCnt: Integer;  
    sResult,sResult2: String;  
begin  
    sResult:=DecodeGroupOrd(sPassPattern,sText);  
    sResult2:="";  
    for iCnt:=1 to (length(sResult) div 3) do  
        sResult2:=sResult2+chr( StrToInt( Copy(sResult,(iCnt-1)*3+1,3) ) );  
    RemoveModel:=sResult2  
end;
```

```
Function RestoreMultipleThree(hxMultipliedThree: TDbIHexa): TOneHundred;  
var byNo1,byNo2: Byte;
```

CrickSpot - Aplicatie Software  
pentru Criptare /P33  
Atestat 2003 - Andrei Neculau  
Colegiul National Vasile  
Alecsandri, Galati

```
    ohResult: TOneHundred;
begin
  byNo1:=StrToInt(hxMultipliedThree[1]);
  Delete(hxMultipliedThree,1,1);
  byNo2:=StrToInt(hxMultipliedThree);
  ohResult:=byNo1+(byNo2-Round(Power(2,byNo1)))*3;
  RestoreMultipleThree:=ohResult;
end;
```

```
Function CSHeadStringDec(sString: String): String;
var sResult,sStr: String;
    iCnt: Integer;
begin
  sResult:="";
  for iCnt:=1 to length(sString) do
    if iCnt mod 3=1 then
      begin
        sStr:=sString[iCnt]+sString[iCnt+1]+sString[iCnt+2];
        sResult:=sResult+chr(RestoreMultipleThree(sStr)+45);
      end;
  CSHeadStringDec:=sResult;
end;
```

```
Procedure HeadGetFileNameSizeDate(var Header: String; var fileList: TCSFileList;
Curr: Integer);
var sStr: String;
    iCnt: Integer;
begin
  sStr:="";
  iCnt:=0;
  repeat
    inc(iCnt);
    if Header[iCnt] in ['0'..'9'] then sStr:=sStr+Header[iCnt];
  until Header[iCnt]='d';
  fileList.Names[Curr]:=CSHeadStringDec(sStr);
  sStr:="";
  repeat
    inc(iCnt);
    if Header[iCnt] in ['0'..'9'] then sStr:=sStr+Header[iCnt];
  until Header[iCnt]='t';
  fileList.Dates[Curr]:=CSHeadStringDec(sStr);
```

CrickSpot - Aplicatie Software  
pentru Criptare /P34  
Atestat 2003 - Andrei Neculau  
Colegiul National Vasile  
Alecsandri, Galati

```

sStr:="";
repeat
  inc(iCnt);
  if Header[iCnt] in ['0'..'9'] then sStr:=sStr+Header[iCnt];
until Header[iCnt]='i';
fileList.Times[Curr]:=CSHeadStringDec(sStr);
sStr:="";
repeat
  inc(iCnt);
  if Header[iCnt] in ['0'..'9'] then sStr:=sStr+Header[iCnt];
until Header[iCnt]='c';
fileList.Sizes.Init[Curr]:=StrToInt(CSHeadStringDec(sStr));
sStr:="";
repeat
  inc(iCnt);
  if Header[iCnt] in ['0'..'9'] then sStr:=sStr+Header[iCnt];
until Header[iCnt]='s';
fileList.Sizes.Code[Curr]:=StrToInt(CSHeadStringDec(sStr));
sStr:="";
repeat
  inc(iCnt);
  if Header[iCnt] in ['0'..'9'] then sStr:=sStr+Header[iCnt];
until Header[iCnt]='|';
fileList.Sizes.Check[Curr]:=StrToInt(CSHeadStringDec(sStr));
end;

```

```

Function CSReadHeader(Header: String; var fileList: TCSFileList): String;
var iCnt,Curr,iCnt1,lastIndex: Integer;
    sResult,sStr1,tempHeader: String;
    bTemp: Boolean;
begin
fileList.NoFiles:=0;
if Pos(Header1,Header)=1 then Delete(Header,1,length(Header1))
else begin CSReadHeader:=""; exit; end;
if Pos(Header2,Header)=1 then Delete(Header,1,length(Header2))
else begin CSReadHeader:=""; exit; end;
sResult:="";
iCnt:=1;
while (iCnt<length(Header)) and (Header[iCnt]<>#250) do begin
  sResult:=sResult+Header[iCnt]; inc(iCnt);
end;

```

CrickSpot - Aplicatie Software  
pentru Criptare /P35  
Atestat 2003 - Andrei Neculau  
Colegiul National Vasile  
Alecsandri, Galati

```

if Header[iCnt]=#250 then Delete(Header,1,length(sResult)+1)
else begin CSReadHeader:=""; exit; end;
if Pos(Header3,Header)=1 then Delete(Header,1,length(Header3))
else begin CSReadHeader:=""; exit; end;
iCnt:=1;
if Header[iCnt]='|' then Delete(Header,1,1)
else begin CSReadHeader:=""; exit; end;
Curr:=0;
iCnt1:=1;
lastIndex:=1;
repeat
  repeat
    bTemp:=false;
    iCnt:=0;
    repeat
      if Header[iCnt1]<>'P' then sStr1:=sStr1+Header[iCnt1]
        else bTemp:=true;
      if not bTemp then
        if not (Header[iCnt1] in ['0'..'9']) then
          if (Header[iCnt1]='d') and (iCnt=0) then iCnt:=1
          else if (Header[iCnt1]='t') and (iCnt=1) then iCnt:=2
            else if (Header[iCnt1]='i') and (iCnt=2) then iCnt:=3
              else if (Header[iCnt1]='c') and (iCnt=3) then iCnt:=4
                else if (Header[iCnt1]='s') and (iCnt=4) then iCnt:=5
                  else if Header[iCnt1]='|' then bTemp:=true;
            inc(iCnt1);
          until bTemp;
        if (iCnt<>5) and (Header[iCnt1]<>'P') then begin CSReadHeader:=""; exit; end
        else if Header[iCnt1]<>'P' then
          begin
            Curr:=Curr+1;
            tempHeader:=Copy(Header,lastIndex,iCnt1-1);
            HeadGetFileNameSizeDate(tempHeader,fileList,Curr);
            inc(fileList.NoFiles);
            sStr1:="";
            bTemp:=false;
            lastIndex:=iCnt1;
          end;
        until (iCnt1>=length(Header));
      CSReadHeader:=sResult;
    end;

```

CrickSpot - Aplicatie Software  
 pentru Criptare /P36  
 Atestat 2003 - Andrei Neculau  
 Colegiul National Vasile  
 Alecsandri, Galati

```
Function IsPlain(Filename: String): Boolean;  
var tempFile: TFileStream;  
    buff: array[1..2] of char;  
begin  
    try  
        tempFile:=TFileStream.Create(Filename,fmOpenRead);  
        tempFile.ReadBuffer(buff,2);  
        if (buff[1]='C') and (buff[2]='S') then IsPlain:=True  
        else IsPlain:=False;  
        FreeAndNil(tempFile);  
    except  
        FreeAndNil(tempFile);  
        IsPlain:=False;  
    end;  
end;
```

```
Function IsBzip(Filename: String): Boolean;  
var tempFile: TFileStream;  
    buff: array[1..2] of char;  
begin  
    try  
        tempFile:=TFileStream.Create(Filename,fmOpenRead);  
        tempFile.ReadBuffer(buff,2);  
        if (buff[1]='B') and (buff[2]='Z') then IsBzip:=True  
        else IsBzip:=False;  
        FreeAndNil(tempFile);  
    except  
        FreeAndNil(tempFile);  
        IsBzip:=False;  
    end;  
end;
```

CrickSpot - Aplicatie Software  
pentru Criptare /P37  
Atestat 2003 - Andrei Neculau  
Colegiul National Vasile  
Alecsandri, Galati