

# Programming In The Past

John Eletto

October 5, 2018

# FORTRAN

Hours Spent: ~3

## FORTRAN Diary

Today I'm starting FORTRAN. It wasn't too hard to install the compiler. I'm using gfortran on my mac.

I'm starting to write FORTRAN now. Who the hell came up with this this? I'd be better off programming on punch cards.

This isn't too bad. Once you figure out how to set variables and make sub-routines, it's a breeze. Somewhat similar to Visual Basic. Now I have to figure out how to do Caesar Cipher.

I learned how to do Caesar Cipher from the internet. You have to use modulo. Now let's do it in FORTRAN.  $(\text{asciiValue} - 65 + \text{shiftAmount}) \bmod 65$

Decrypt is just the same thing except minus the shiftAmount. Done.

## FORTRAN Code

```
! file: CaesarCipher.f90
! author: John Eletto
! website: johneletto.com
! github: git.johneletto.com

program CaesarCipher

    ! set key for cipher
    INTEGER :: shiftAmount = 26

    ! set string to encrypt
    CHARACTER(len = 38) :: word = "THIS IS A TEST STRING FROM JOHN ELETTO"

    ! call encrypt
    call encrypt(word, shiftAmount)

    ! call decrypt
    call decrypt(word, shiftAmount)

end program CaesarCipher

! Encrypt SubRoutine
subroutine encrypt(word, shiftAmount)
```

```

! declaring needed variables
CHARACTER(*) :: word
INTEGER :: shiftAmount
INTEGER :: i

! loop for every character of our string
do i = 1, len(word)
    select case(word(i:i))
        ! if the character is A-Z
        case ("A" : "Z")
            ! perform caesar cipher on the current character
            ! achar returns the character value from ASCII Number sequence
            ! iachar retrns the ASCII number from a character
            word(i:i) = achar(modulo(iachar(word(i:i)) - 65 + shiftAmount, 26) + 65)
        ! if the character is a space, preserve the space
        case (" ")
            word(i:i) = " "
    end select
end do

print *, "Encrypted: ", word
end subroutine encrypt

! Decrypt SubRoutine
subroutine decrypt(word, shiftAmount)
    ! Declare needed variables
    CHARACTER(*) :: word
    INTEGER :: shiftAmount
    INTEGER :: i
    INTEGER :: j

    ! loop from 1 to shiftAmount (this gives all possible combinations)
    do j = 1, shiftAmount
        ! loop for every character of our string
        do i = 1, len(word)
            ! select current character
            select case(word(i:i))
                ! if character is A-Z
                case ("A" : "Z")
                    word(i:i) = achar(modulo(iachar(word(i:i)) - 65 - shiftAmount, 26) + 65)
                ! Preserve spaces
                case (" ")
                    word(i:i) = " "
            end select
        end do
        ! Print current and then decrement shiftAmount and do it again.

```

```

        print *, "Caesar ", shiftAmount, ": ", word
        shiftAmount = shiftAmount - 1
    end do

```

```

end subroutine decrypt

```

## FORTRAN Output

```

Encrypted: THIS IS A TEST STRING FROM JOHN ELETTO
Caesar      26 : THIS IS A TEST STRING FROM JOHN ELETTO
Caesar      25 : UIJT JT B UFTU TUSJOH GSPN KPIO FMFUUP
Caesar      24 : WKLV LV D WHWV VWULQJ IURP MRKQ HOHWWR
Caesar      23 : ZNOY OY G ZKYZ YZXOTM LXUS PUNT KRKZZU
Caesar      22 : DRSC SC K DOCD CDBSXQ PBWY TYRX OVODDY
Caesar      21 : IWXH XH P ITHI HIGXCV UGDB YDWC TATIID
Caesar      20 : OCDN DN V OZNO NOMDIB AMJH EJCI ZGZOOJ
Caesar      19 : VJKU KU C VGUU UVTKPI HTQO LQJP GNGVVQ
Caesar      18 : DRSC SC K DOCD CDBSXQ PBWY TYRX OVODDY
Caesar      17 : MABL BL T MXLM LMKBGZ YKHF CHAG XEXMMH
Caesar      16 : WKLV LV D WHWV VWULQJ IURP MRKQ HOHWWR
Caesar      15 : HWVG WG O HSGH GHFWBU TFCA XCVB SZSHHC
Caesar      14 : THIS IS A TEST STRING FROM JOHN ELETTO
Caesar      13 : GUVF VF N GRFG FGEVAT SEBZ WBUA RYRGGB
Caesar      12 : UIJT JT B UFTU TUSJOH GSPN KPIO FMFUUP
Caesar      11 : JXYI YI Q JUIJ LJHYDW VHEC ZEXD UBUJJE
Caesar      10 : ZNOY OY G ZKYZ YZXOTM LXUS PUNT KRKZZU
Caesar       9 : QEFP FP X QBPQ PQOFKD COLJ GLEK BIBQQL
Caesar       8 : IWXH XH P ITHI HIGXCV UGDB YDWC TATIID
Caesar       7 : BPQA QA I BMAB ABZQVO NZWU RWPV MIMBBW
Caesar       6 : VJKU KU C VGUU UVTKPI HTQO LQJP GNGVVQ
Caesar       5 : QEFP FP X QBPQ PQOFKD COLJ GLEK BIBQQL
Caesar       4 : MABL BL T MXLM LMKBGZ YKHF CHAG XEXMMH
Caesar       3 : JXYI YI Q JUIJ LJHYDW VHEC ZEXD UBUJJE
Caesar       2 : HWVG WG O HSGH GHFWBU TFCA XCVB SZSHHC
Caesar       1 : GUVF VF N GRFG FGEVAT SEBZ WBUA RYRGGB

```

```

Encrypted: XLMW MW E XIWX WXVMRK JVSQ NSLR IPIXXS
Caesar      4 : THIS IS A TEST STRING FROM JOHN ELETTO
Caesar      3 : QEFP FP X QBPQ PQOFKD COLJ GLEK BIBQQL
Caesar      2 : OCDN DN V OZNO NOMDIB AMJH EJCI ZGZOOJ
Caesar      1 : NBCM CM U NYMN MNLCHA ZLIG DIBH YFYNNI

```

## COBOL

Hours Spent:  $\infty$

## COBOL Diary

COBOL looks like an absolute shit show. Saving this for last like a true procrastinator.

COBOL is impossible. I don't know why anyone would try to make a programming language for business people. This is what you get. I have absolutely no clue what's going on in COBOL and I'm not really interested in figuring it out. Skipping COBOL.

## COBOL Code

### **Cobol Programmer**



## BASIC

Hours Spent: ~2

## BASIC Diary

Basic isn't very hard. I have some experience with Visual Basic and this isn't very different.

## BASIC Code

```
' file: CaesarCipher.bas
' author: John Eletto
' website: johneletto.com
' github: git.johneletto.com

' Subroutine for encrypting with the Caesar Cipher
Sub encrypt(word As String, shiftAmount as Integer)
    Dim currentCharAscii As Integer
    For i As Integer = 0 To Len(word)
```

```

        currentCharAscii = word[i]
        Select Case As Const word[i]
            ' If character is 'A' to 'Z'
            Case 65 To 90
                word[i] = currentCharAscii + shiftAmount
                If word[i] > 90 Then word[i] -= 26
            ' If character is a space
            Case 32
                word[i] = word[i]
        End Select

    Next
    Print "Encrypted: "; word
End Sub

' Subroutine for decrypting with the Caesar Cipher
Sub decrypt(word As String, shiftAmount as Integer)
    Dim CurrentCharAscii As Integer
    'For i As Integer = 0 to shiftAmount
        For j As Integer = 0 to Len(word)
            currentCharAscii = word[j]
            Select Case As Const word[j]
                Case 65 To 90
                    word[j] = currentCharAscii - shiftAmount
                    If word[j] < 65 Then word[j] += 26
                Case 32
                    word[j] = word[j]
            End Select
        Next
        Print "Decrypted: "; ": "; word
        'If shiftAmount = 1 Then return
        'decrypt word, shiftAmount - 1
    'Next
End Sub

' Subroutine for solving with the Caesar Cipher
Sub solve(word As String, shiftAmount as Integer)
    Dim CurrentCharAscii As Integer
    For j As Integer = 0 to Len(word)
        currentCharAscii = word[j]
        Select Case As Const word[j]
            Case 65 To 90
                word[j] = currentCharAscii - shiftAmount
                If word[j] < 65 Then word[j] += 26
            Case 32
                word[j] = word[j]

```

```

        End Select
    Next
    Print "Caesar "; shiftAmount; ": "; word
    If shiftAmount <= 1 Then return
    solve word, shiftAmount - 1
End Sub

```

```

Dim As Integer shiftAmount = 25
Dim As String word = "THIS IS A TEST STRING FROM JOHN ELETTO"

```

```

encrypt word, shiftAmount

```

```

decrypt word, shiftAmount

```

```

Print "Solving:"

```

```

solve word, 26

```

## BASIC Output

```

Encrypted: SGHR HR Z SDRS RSQHMF EQNL INGM DKDSSN
Decrypted: : THIS IS A TEST STRING FROM JOHN ELETTO
Solving:
Caesar 26: THIS IS A TEST STRING FROM JOHN ELETTO
Caesar 25: UIJT JT B UFTU TUSJOH GSPN KPIO FMFUUP
Caesar 24: WKLV LV D WHWV VWULQJ IURP MRKQ HOHWWR
Caesar 23: ZNOY OY G ZKYZ YZXOTM LXUS PUNT KRKZZU
Caesar 22: DRSC SC K DOCD CDBSXQ PBYW TYRX OVODDY
Caesar 21: IWXH XH P ITHI HIGXCV UGDB YDWC TATIID
Caesar 20: OCDN DN V OZNO NOMDIB AMJH EJCI ZGZOOJ
Caesar 19: VJKU KU C VGUU UVTKPI HTQO LQJP GNGVVQ
Caesar 18: DRSC SC K DOCD CDBSXQ PBYW TYRX OVODDY
Caesar 17: MABL BL T MXLM LMKBGZ YKHF CHAG XEXMMH
Caesar 16: WKLV LV D WHWV VWULQJ IURP MRKQ HOHWWR
Caesar 15: HWVG WG O HSGH GHFWBU TFCA XCVB SZSHHC
Caesar 14: THIS IS A TEST STRING FROM JOHN ELETTO
Caesar 13: GUVF VF N GRFG FGEVAT SEBZ WBUA RYRGGB
Caesar 12: UIJT JT B UFTU TUSJOH GSPN KPIO FMFUUP
Caesar 11: JXYI YI Q JUIJ IJHYDW VHEC ZEXD UBUJJE
Caesar 10: ZNOY OY G ZKYZ YZXOTM LXUS PUNT KRKZZU
Caesar 9: QEFP FP X QBPQ PQOFKD COLJ GLEK BIBQQL
Caesar 8: IWXH XH P ITHI HIGXCV UGDB YDWC TATIID
Caesar 7: BPQA QA I BMAB ABZQVO NZWU RWPV MIMBBW
Caesar 6: VJKU KU C VGUU UVTKPI HTQO LQJP GNGVVQ

```

```

Caesar 5: QEFP FP X QBPO PQOFKD COLJ GLEK BIBQQL
Caesar 4: MABL BL T MXLM LMKBGZ YKHF CHAG XEXMMH
Caesar 3: JXYI YI Q JUIJ LJHYDW VHEC ZEXD UBUJJE
Caesar 2: HVWG WG O HSGH GHFWBU TFCA XCVB SZSHHC
Caesar 1: GUVF VF N GRFG FGEVAT SEBZ WBUA RYRGGB

```

```

Encrypted: WKLV LV D WHWV VWULQJ IURP MRKQ HOHWWR
Decrypted: : THIS IS A TEST STRING FROM JOHN ELETTO
Solving:

```

```

Caesar 26: THIS IS A TEST STRING FROM JOHN ELETTO
Caesar 25: UIJT JT B UFTU TUSJOH GSPN KPIO FMFUUP
Caesar 24: WKLV LV D WHWV VWULQJ IURP MRKQ HOHWWR
Caesar 23: ZNOY OY G ZKYZ YZXOTM LXUS PUNT KRKZZU
Caesar 22: DRSC SC K DOCD CDBSXQ PBYW TYRX OVODDY
Caesar 21: IWXH XH P ITHI HIGXCV UGDB YDWC TATIID
Caesar 20: OCDN DN V OZNO NOMDIB AMJH EJCI ZGZOOJ
Caesar 19: VJKU KU C VGUV UVTKPI HTQO LQJP GNGVVQ
Caesar 18: DRSC SC K DOCD CDBSXQ PBYW TYRX OVODDY
Caesar 17: MABL BL T MXLM LMKBGZ YKHF CHAG XEXMMH
Caesar 16: WKLV LV D WHWV VWULQJ IURP MRKQ HOHWWR
Caesar 15: HVWG WG O HSGH GHFWBU TFCA XCVB SZSHHC
Caesar 14: THIS IS A TEST STRING FROM JOHN ELETTO
Caesar 13: GUVF VF N GRFG FGEVAT SEBZ WBUA RYRGGB
Caesar 12: UIJT JT B UFTU TUSJOH GSPN KPIO FMFUUP
Caesar 11: JXYI YI Q JUIJ LJHYDW VHEC ZEXD UBUJJE
Caesar 10: ZNOY OY G ZKYZ YZXOTM LXUS PUNT KRKZZU
Caesar 9: QEFP FP X QBPO PQOFKD COLJ GLEK BIBQQL
Caesar 8: IWXH XH P ITHI HIGXCV UGDB YDWC TATIID
Caesar 7: BPQA QA I BMAB ABZQVO NZWU RWPV MIMBBW
Caesar 6: VJKU KU C VGUV UVTKPI HTQO LQJP GNGVVQ
Caesar 5: QEFP FP X QBPO PQOFKD COLJ GLEK BIBQQL
Caesar 4: MABL BL T MXLM LMKBGZ YKHF CHAG XEXMMH
Caesar 3: JXYI YI Q JUIJ LJHYDW VHEC ZEXD UBUJJE
Caesar 2: HVWG WG O HSGH GHFWBU TFCA XCVB SZSHHC
Caesar 1: GUVF VF N GRFG FGEVAT SEBZ WBUA RYRGGB

```

## Pascal

Hours Spent: ~2

### Pascal Diary

Pascal isn't too bad. The syntax is a little weird, and the comments are really weird. However, at least it makes logical sense (unlike COBOL).



## Pascal Code

```
{
    file : CaesarCipher.pas
    author : John Eletto
    website : johneletto.com
    github : git.johneletto.com
}

Program CaesarCipher(output);

{ ProcEDURE to encrypt a word given a word and shiftAmount }
procedure encrypt(var word: string; shiftAmount: integer);
var
    i: integer;
begin
    for i := 1 to length(word) do
        case word[i] of
            'A'..'Z': word[i] := chr(ord('A') + (ord(word[i]) - ord('A') + s
            ' ': word[i] := ' ';
        end;
        writeln('Encrypted: ', word);
    end;

{ ProcEDURE to decrypt a word given a word and shiftAmount }
procedure decrypt(var word: string; shiftAmount: integer);
var
    i: integer;
begin
    for i := 1 to length(word) do
        case word[i] of
            'A'..'Z': word[i] := chr(ord('A') + (ord(word[i]) - ord('A') - s
            ' ': word[i] := ' ';
        end;
        writeln('Decrypted: ', word);
    end;

{ ProcEDURE to solve a caesar cipher by returning all solutions }
procedure solve(var word: string; shiftAmount: integer);
var
    i: integer;
begin
    for i := 1 to length(word) do
        case word[i] of
            'A'..'Z': word[i] := chr(ord('A') + (ord(word[i]) - ord('A') - s
            ' ': word[i] := ' ';
```

```

        end;
        writeln('Caesar ', shiftAmount, ': ', word);
        if (shiftAmount = 1)
        then
        else solve(word, shiftAmount - 1);

    end;

var
    word: string;
    shiftAmount: integer;

begin
    shiftAmount := 15;
    word := 'THIS IS A TEST MESSAGE FROM JOHN ELETTO';

    encrypt(word, shiftAmount);
    decrypt(word, shiftAmount);
    writeln('Solving:');
    solve(word, 26);
end.

```

## Pascal Output

Shift: 15

```

Encrypted: IWXH XH P ITHI BTHHPVT UGDB YDWC TATIID
Decrypted: THIS IS A TEST MESSAGE FROM JOHN ELETTO
Solving:
Caesar 26: THIS IS A TEST MESSAGE FROM JOHN ELETTO
Caesar 25: UIJT JT B UFTU NFTTBHF GSPN KPIO FMFUUP
Caesar 24: WKLV LV D WHW PHVVDJH IURP MRKQ HOHWWR
Caesar 23: ZNOY OY G ZKYZ SKYYGMK LXUS PUNT KRKZZU
Caesar 22: DRSC SC K DOCD WOOCKQO PBYW TYRX OVODDY
Caesar 21: IWXH XH P ITHI BTHHPVT UGDB YDWC TATIID
Caesar 20: OCDN DN V OZNO HZNNVBZ AMJH EJCI ZGZOOJ
Caesar 19: VJKU KU C VGUU OGUUCIG HTQO LQJP GNGVVQ
Caesar 18: DRSC SC K DOCD WOOCKQO PBYW TYRX OVODDY
Caesar 17: MABL BL T MXLM FXLLTZX YKHF CHAG XEXMMH
Caesar 16: WKLV LV D WHW PHVVDJH IURP MRKQ HOHWWR
Caesar 15: HWVG WG O HSGH ASGGOUS TFCA XCVB SZSHHC
Caesar 14: THIS IS A TEST MESSAGE FROM JOHN ELETTO
Caesar 13: GUVF VF N GRFG ZRFFNTR SEBZ WBUA RYRGGB
Caesar 12: UIJT JT B UFTU NFTTBHF GSPN KPIO FMFUUP

```

Caesar 11: JXYI YI Q JULJ CUIQWU VHEC ZEXD UBUJJE  
 Caesar 10: ZNOY OY G ZKYZ SKYYGMK LXUS PUNT KRKZZU  
 Caesar 9: QEFP FP X QBPQ JBPPXDB COLJ GLEK BIBQQL  
 Caesar 8: IWXH XH P ITHI BTHHPVT UGDB YDWC TATIID  
 Caesar 7: BPQA QA I BMAB UMAAIOM NZWU RWPV MIMBBW  
 Caesar 6: VJKU KU C VGUU OGUUCIG HTQO LQJP GNGVVQ  
 Caesar 5: QEFP FP X QBPQ JBPPXDB COLJ GLEK BIBQQL  
 Caesar 4: MABL BL T MXLM FXLLTZ XKHF CHAG XEXMMH  
 Caesar 3: JXYI YI Q JULJ CUIQWU VHEC ZEXD UBUJJE  
 Caesar 2: HVWG WG O HSGH ASGGOUS TFCA XCVB SZSHHC  
 Caesar 1: GUVF VF N GRFG ZRFFNTR SEBZ WBUA RYRGGB

Shift: 7

Encrypted: AOPZ PZ H ALZA TLZZHNL MYVT QVOU LSLAAV  
 Decrypted: THIS IS A TEST MESSAGE FROM JOHN ELETTO  
 Solving:

Caesar 26: THIS IS A TEST MESSAGE FROM JOHN ELETTO  
 Caesar 25: UIJT JT B UFTU NFFTBFH GSPN KPIO FMFUUP  
 Caesar 24: WKLV LV D WHWV PHVVDJH IURP MRKQ HOHWWR  
 Caesar 23: ZNOY OY G ZKYZ SKYYGMK LXUS PUNT KRKZZU  
 Caesar 22: DRSC SC K DOCD WOCCCKQO PBYW TYRX OVODDY  
 Caesar 21: IWXH XH P ITHI BTHHPVT UGDB YDWC TATIID  
 Caesar 20: OCDN DN V OZNO HZNNVBZ AMJH EJCI ZGZOOJ  
 Caesar 19: VJKU KU C VGUU OGUUCIG HTQO LQJP GNGVVQ  
 Caesar 18: DRSC SC K DOCD WOCCCKQO PBYW TYRX OVODDY  
 Caesar 17: MABL BL T MXLM FXLLTZ XKHF CHAG XEXMMH  
 Caesar 16: WKLV LV D WHWV PHVVDJH IURP MRKQ HOHWWR  
 Caesar 15: HVWG WG O HSGH ASGGOUS TFCA XCVB SZSHHC  
 Caesar 14: THIS IS A TEST MESSAGE FROM JOHN ELETTO  
 Caesar 13: GUVF VF N GRFG ZRFFNTR SEBZ WBUA RYRGGB  
 Caesar 12: UIJT JT B UFTU NFFTBFH GSPN KPIO FMFUUP  
 Caesar 11: JXYI YI Q JULJ CUIQWU VHEC ZEXD UBUJJE  
 Caesar 10: ZNOY OY G ZKYZ SKYYGMK LXUS PUNT KRKZZU  
 Caesar 9: QEFP FP X QBPQ JBPPXDB COLJ GLEK BIBQQL  
 Caesar 8: IWXH XH P ITHI BTHHPVT UGDB YDWC TATIID  
 Caesar 7: BPQA QA I BMAB UMAAIOM NZWU RWPV MIMBBW  
 Caesar 6: VJKU KU C VGUU OGUUCIG HTQO LQJP GNGVVQ  
 Caesar 5: QEFP FP X QBPQ JBPPXDB COLJ GLEK BIBQQL  
 Caesar 4: MABL BL T MXLM FXLLTZ XKHF CHAG XEXMMH  
 Caesar 3: JXYI YI Q JULJ CUIQWU VHEC ZEXD UBUJJE  
 Caesar 2: HVWG WG O HSGH ASGGOUS TFCA XCVB SZSHHC  
 Caesar 1: GUVF VF N GRFG ZRFFNTR SEBZ WBUA RYRGGB

## Scala

Hours Spent: ~2

### Scala Diary

Scala is cool. It feels like I'm back in 2018. Thanks Scala.

### Scala Code

```
// file: CaesarCipher.pas
// author: John Eletto
// website: johneletto.com
// github: git.johneletto.com

object CaesarCipher{
  var word = "THIS IS A TEST MESSAGE FROM JOHN ELETTO";
  var shiftAmount = 15;
  val alpha = 'A' to 'Z'

  def main(args: Array[String]) {
    encrypt();
    decrypt();
    println("Solving:");
    solve(26);
  }

  def encrypt(){
    val encrypted = word.map(c =>
      if(c == ' '){
        c
      }
      else {
        alpha((c - 'A' + shiftAmount + 26) % 26)
      }
    )
    word = encrypted;
    println("Encrypted: " + word);
  }

  def decrypt(){
    val decrypted = word.map(c =>
      if(c == ' '){
        c
      }
      else {
        alpha((c - 'A' - shiftAmount + 26) % 26)
      }
    )
    word = decrypted;
    println("Decrypted: " + word);
  }
}
```

```

    }
  )

  word = decrypted;
  println("Decrypted: " + word);
}

def solve(maxShift: Integer){
  val decrypted = word.map(c =>
    if(c == ' '){
      c
    } else {
      alpha((c - 'A' - maxShift + 26) % 26)
    }
  )

  word = decrypted;
  println("Caesar " + maxShift + ": " + word);

  if(maxShift <= 1){
    return;
  }

  solve(maxShift - 1);
}
}

```

## Scala Output

Shift: 15

```

Encrypted: IWXH XH P ITHI BTHHPVT UGDB YDWC TATIID
Decrypted: THIS IS A TEST MESSAGE FROM JOHN ELETTO
Solving:
Caesar 26: THIS IS A TEST MESSAGE FROM JOHN ELETTO
Caesar 25: UIJT JT B UFTU NFTTBHF GSPN KPIO FMFUUP
Caesar 24: WKLV LV D WHW PHVVDJH IURP MRKQ HOHWWR
Caesar 23: ZNOY OY G ZKYZ SKYYGMK LXUS PUNT KRKZZU
Caesar 22: DRSC SC K DOCD WOOCKQO PBYW TYRX OVODDY
Caesar 21: IWXH XH P ITHI BTHHPVT UGDB YDWC TATIID
Caesar 20: OCDN DN V OZNO HZNNVBZ AMJH EJCI ZGZOOJ
Caesar 19: VJKU KU C VGUU OGUUCIG HTQO LQJP GNGVVQ
Caesar 18: DRSC SC K DOCD WOOCKQO PBYW TYRX OVODDY
Caesar 17: MABL BL T MXLM FXLLTZX YKHF CHAG XEXMMH
Caesar 16: WKLV LV D WHW PHVVDJH IURP MRKQ HOHWWR
Caesar 15: HWVG WG O HSGH ASGGOUS TFCA XCVB SZSHHC

```

Caesar 14: THIS IS A TEST MESSAGE FROM JOHN ELETTO  
 Caesar 13: GUVF VF N GRFG ZRFFNTR SEBZ WBUA RYRGGB  
 Caesar 12: UIJT JT B UFTU NFTTBHF GSPN KPIO FMFUUP  
 Caesar 11: JXYI YI Q JUIJ CUIQWU VHEC ZEXD UBUJJE  
 Caesar 10: ZNOY OY G ZKYZ SKYYGMK LXUS PUNT KRKZZU  
 Caesar 9: QEFP FP X QBPQ JBPPXDB COLJ GLEK BIBQQL  
 Caesar 8: IWXH XH P ITHI BTHHPVT UGDB YDWC TATIID  
 Caesar 7: BPQA QA I BMAB UMAAIOM NZWU RWPV MIMBBW  
 Caesar 6: VJKU KU C VGUU OGUUCIG HTQO LQJP GNGVVQ  
 Caesar 5: QEFP FP X QBPQ JBPPXDB COLJ GLEK BIBQQL  
 Caesar 4: MABL BL T MXLM FXLLTZ XKHF CHAG XEXMMH  
 Caesar 3: JXYI YI Q JUIJ CUIQWU VHEC ZEXD UBUJJE  
 Caesar 2: HVWG WG O HSGH ASGGOUS TFCA XCVB SZSHHC  
 Caesar 1: GUVF VF N GRFG ZRFFNTR SEBZ WBUA RYRGGB

Shift: 21

Encrypted: OCDN DN V OZNO HZNNVBZ AMJH EJCI ZGZOOJ  
 Decrypted: THIS IS A TEST MESSAGE FROM JOHN ELETTO  
 Solving:

Caesar 26: THIS IS A TEST MESSAGE FROM JOHN ELETTO  
 Caesar 25: UIJT JT B UFTU NFTTBHF GSPN KPIO FMFUUP  
 Caesar 24: WKLV LV D WHWV PHVVDJH IURP MRKQ HOHWWR  
 Caesar 23: ZNOY OY G ZKYZ SKYYGMK LXUS PUNT KRKZZU  
 Caesar 22: DRSC SC K DOCD WOCKQO PBYW TYRX OVODDY  
 Caesar 21: IWXH XH P ITHI BTHHPVT UGDB YDWC TATIID  
 Caesar 20: OCDN DN V OZNO HZNNVBZ AMJH EJCI ZGZOOJ  
 Caesar 19: VJKU KU C VGUU OGUUCIG HTQO LQJP GNGVVQ  
 Caesar 18: DRSC SC K DOCD WOCKQO PBYW TYRX OVODDY  
 Caesar 17: MABL BL T MXLM FXLLTZ XKHF CHAG XEXMMH  
 Caesar 16: WKLV LV D WHWV PHVVDJH IURP MRKQ HOHWWR  
 Caesar 15: HVWG WG O HSGH ASGGOUS TFCA XCVB SZSHHC  
 Caesar 14: THIS IS A TEST MESSAGE FROM JOHN ELETTO  
 Caesar 13: GUVF VF N GRFG ZRFFNTR SEBZ WBUA RYRGGB  
 Caesar 12: UIJT JT B UFTU NFTTBHF GSPN KPIO FMFUUP  
 Caesar 11: JXYI YI Q JUIJ CUIQWU VHEC ZEXD UBUJJE  
 Caesar 10: ZNOY OY G ZKYZ SKYYGMK LXUS PUNT KRKZZU  
 Caesar 9: QEFP FP X QBPQ JBPPXDB COLJ GLEK BIBQQL  
 Caesar 8: IWXH XH P ITHI BTHHPVT UGDB YDWC TATIID  
 Caesar 7: BPQA QA I BMAB UMAAIOM NZWU RWPV MIMBBW  
 Caesar 6: VJKU KU C VGUU OGUUCIG HTQO LQJP GNGVVQ  
 Caesar 5: QEFP FP X QBPQ JBPPXDB COLJ GLEK BIBQQL  
 Caesar 4: MABL BL T MXLM FXLLTZ XKHF CHAG XEXMMH  
 Caesar 3: JXYI YI Q JUIJ CUIQWU VHEC ZEXD UBUJJE  
 Caesar 2: HVWG WG O HSGH ASGGOUS TFCA XCVB SZSHHC

Caesar 1: GUVF VF N GRFG ZRFFNTR SEBZ WBUA RYRGB