Programming In The Past

John Eletto

October 5, 2018

FORTRAN

Hours Spent: ∼3

FORTRAN Diary

Dear Diary,
Today I'm starting FORTRAN. It wasn't to hard to install the compiler. I'm using gfortran on my mac.
Love, John

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

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

Dear Diary,
I learned how to do Caesar Cipher from the internet. You have to use modulo. Now let's do it in FORTRAN. (asciiValue - 65 + shiftAmount) mod 65
Love, John

Dear Diary,
Decrypt is just the same thing except minus the shiftAmount. Done.
Love, John

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, 2
            !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 - shiftAmoun
                ! 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 WHVW 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 WHVW 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 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 VGUV 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 IJHYDW VHEC ZEXD UBUJJE
 Caesar           2 : HVWG WG O HSGH GHFWBU TFCA XCVB SZSHHC
 Caesar           1 : GUVF VF N GRFG FGEVAT SEBZ WBUA RYRGGB
```

COBOL

Hours Spent: ∼

COBOL Diary

Dear Diary,
COBOL looks like an absolute shit show. Saving this for last like a true procrastinator.
Love, John

COBOL Code

```
Insert Code Here
```

BASIC

Hours Spent: ∼

BASIC Diary

Insert Diary Here

BASIC Code

```
Insert Code Here
```

BASIC Code

```
Insert Code Here
```

Pascal

Hours Spent: ∼

0.0.1 Pascal Diary

Insert Diary Here

Pascal Code

```
 Insert Code Here
```

Scala

Hours Spent: ∼

Scala Diary

Insert Diary Here

Scala Code

```
Insert Code Here
```