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
```

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: ∼

0.0.1 BASIC Diary

Insert Diary Here

BASIC Code

```
Insert Code Here
```

BASIC Code

```
Insert Code Here
```

Pascal

Hours Spent: $\sim$

0.0.2 Pascal Diary

Insert Diary Here

Pascal Code

```
Insert Code Here
```

Scala

Hours Spent: $\sim$

0.0.3 Scala Diary

Insert Diary Here

Scala Code

```
Insert Code Here
```