Enigma encoding pseudocode

Slot1, Slot2, Slot3 = Rotor Class

Reflector = Reflector Class

Plugboard = Plugboard Class

Accept Input letter

Force letter to uppercase

SwitchedLetter = Plugboard.Select(Input Letter)

Slot1.Rotate() *--Note that this should increment Rotor1.step*

If Slot2.DoubleRotate

Slot2.Rotate()

Rotate.DoubleRotate = False

Else If Slot1.IsNotched *-Note that this will only handle 3 rotors. Need recursive routine to handle N rotors*

Slot2.Rotate() *--Note that this should increment Rotor2.step*

Rotate.DoubleRotate=True

If Slot2.IsNotched

Slot3.Rotate() *--Note that this should increment Rotor3.step*

End

End

EncodedLetter = Rotor1Array[ ASC(SwitchedLetter) – 65 ] --Rotor arrays should be 0 based

EncodedLetter = Rotor2Array[ ASC(EncodedLetter) – 65 – Rotor1.Step ] --If ArrayPosition < 0, Add 26

EncodedLetter = Rotor3Array[ ASC(EncodedLetter) – 65 – Rotor2.Step ] --If ArrayPosition < 0, Add 26

EncodedLetter = ReflectorArray[ ASC(EncodedLetter) – 65 – Rotor3.Step] --If ArrayPosition < 0, Add 26

EncodedLetter = CHR ( ASC(EncodedLetter) – 65 + Rotor3.Step )

EncodedLetter = CHR (Rotor1Array.Find (CHR( Rotor2.Find (CHR(Rotor3.Find(EncodedLetter) - Rotor3.Step + Rotor2.Step) ) - Rotor2.Step + Rotor1.Step)) – Rotor1.Step)