**Breaking the Enigma Code**

**Testing results for functions:**

**steckerAdd:**

**\*Bombe\_Testing>** st1

[('L','P'),('C','Q'),('M','R'),('H','S'),('G','T'),('E','U'),('D','V'),('I','W'),('A','X')]

**\*Bombe\_Testing>** let stecker = st1

**\*Bombe\_Testing>** let pairWillWork = ('F','F')

**\*Bombe\_Testing>** let pairWontWork = ('A','B')

**\*Bombe\_Testing>** let pairWontWorkReverse = ('B','A')

This tests show that if the letters in the pair aren’t already in the stecker list, they get added and returned.

**\*Bombe\_Testing>** steckerAdd pairWillWork stecker

Just [('L','P'),('C','Q'),('M','R'),('H','S'),('G','T'),('E','U'),('D','V'),('I','W'),('A','X'),('F','F')]

This test shows that if the letters in the pair are already in the stecker list, then Nothing is returned

**\*Bombe\_Testing>** steckerAdd pairWontWork stecker

Nothing

This test shows that the previous example works no matter what way round the letters are in the pair.

**\*Bombe\_Testing>** steckerAdd pairWontWorkReverse stecker

Nothing

This test shows that steckerAdd returns the stecker list if the pair it tries to add is already in the list.

**\*Bombe\_Testing>** steckerAdd ('A','X') stecker

Just [('L','P'),('C','Q'),('M','R'),('H','S'),('G','T'),('E','U'),('D','V'),('I','W'),('A','X')]

This test shows that the previous example works no matter what way round the letters are in the pair.

**\*Bombe\_Testing>** steckerAdd ('X','A') stecker

Just [('L','P'),('C','Q'),('M','R'),('H','S'),('G','T'),('E','U'),('D','V'),('I','W'),('A','X')]

**followMenu:**

**\*Bombe\_Testing>** c1

[('A','I'),('I','D'),('D','E'),('E','G'),('G','H'),('H','M'),('M','C'),('C','L')]

This test shows that if a complete stecker list cannot be created using the initial stecker, nothing is returned.

**\*Bombe\_Testing>** followMenu c1 (longestMenu c1) [('A','A')] (0,0,0)

Nothing

This test shows that if the correct initial stecker and offsets is used, a solution is returned.

**\*Bombe\_Testing>** followMenu c1 (longestMenu c1) [('A','X')] (0,0,0)

Just [('A','X'),('W','I'),('V','D'),('U','E'),('T','G'),('S','H'),('R','M'),('Q','C'),('P','L')]

This test shows that if the correct initial stecker is used, but the incorrect offsets if used, Nothing is returned.

**\*Bombe\_Testing>** followMenu c1 (longestMenu c1) [('A','X')] (0,0,1)

Nothing

**findStecker:**

This test shows that if the initial stecker doesn’t produce a solution, it goes through the new steckers until one does, or all 26 options have been tried.

**\*Bombe\_Testing>** findStecker c1 (longestMenu c1) [('A','A')] (0,0,0)

Just [('A','X'),('W','I'),('V','D'),('U','E'),('T','G'),('S','H'),('R','M'),('Q','C'),('P','L')]

**\*Bombe\_Testing>** findStecker crib2 (longestMenu crib2) [('T','T')] (0,0,5)

Just [('T','G'),('C','E'),('K','F'),('D','S'),('R','H'),('O','L'),('N','N'),('B','V'),('A','U'),('P','M'),('Q','Q')]

This test shows that if findStecker can’t find a solution with the current offsets, Nothing is returned.

**\*Bombe\_Testing>** findStecker crib2 (longestMenu crib2) [('T','T')] (0,0,0)

Nothing

**breakEA:**

These test show that breakEA will find a solution to the crib, given the initial stecker and initial offset of (0,0,0). The function returns both the offset position and the stecker list.

**\*Bombe\_Testing>** breakEA c1 (longestMenu c1) [('A','A')] (0,0,0)

Just ((0,0,0),[('A','X'),('W','I'),('V','D'),('U','E'),('T','G'),('S','H'),('R','M'),('Q','C'),('P','L')])

**\*Bombe\_Testing>** breakEA crib2 (longestMenu crib2) [('T','T')] (0,0,0)

Just ((0,0,5),[('T','G'),('C','E'),('K','F'),('D','S'),('R','H'),('O','L'),('N','N'),('B','V'),('A','U'),('P','M'),('Q','Q')])

**\*Bombe\_Testing>** breakEA crib3 (longestMenu crib3) [('U','U')] (0,0,0)

Just ((1,2,3),[('U','U'),('N','N'),('Y','B'),('V','E'),('T','T'),('I','I'),('L','L'),('O','O'),('Z','A'),('H','H'),('C','X')])

This test shows that if there is no possible solution for the given crib (using the initial stecker corresponding to the menu), then after it goes through all options up to (25,25,25), Nothing is returned.

**\*Bombe\_Testing>** weatherMenu

[('W','R'),('E','W'),('T','I'),('T','V'),('E','T'),('R','Y'),('V','R'),('O','E'),('R','S'),('H','X'),('E','B'),('R','F'),('S','O'),('A','G'),('G','K'),('E','U'),('B','H'),('I','Q'),('S','B'),('K','A'),('A','I'),('Y','S'),('A','E')]

**\*Bombe\_Testing>** breakEA weatherMenu (longestMenu weatherMenu) [('S','S')] (0,0,0)

Nothing

**breakEnigma:**

These tests show that breakEnigma works for all 3 of the tests given, producing a stecker list and the offsets for that point.

**\*Bombe\_Testing>** breakEnigma c1

Just ((0,0,0),[('A','X'),('W','I'),('V','D'),('U','E'),('T','G'),('S','H'),('R','M'),('Q','C'),('P','L')])

**\*Bombe\_Testing>** breakEnigma crib2

Just ((0,0,5),[('T','G'),('C','E'),('K','F'),('D','S'),('R','H'),('O','L'),('N','N'),('B','V'),('A','U'),('P','M'),('Q','Q')])

**\*Bombe\_Testing>** breakEnigma crib3

Just ((1,2,3),[('U','U'),('N','N'),('Y','B'),('V','E'),('T','T'),('I','I'),('L','L'),('O','O'),('Z','A'),('H','H'),('C','X')])

**\*Bombe\_Testing>** breakEnigma weatherMenu

Nothing

These tests show that the results above are correct as they either produce full decrypts of the ciphered message or (in the case of test 2 & 3) a partial decode of the cipher that is easy to read.

**\*Bombe\_Testing>** let stecker1 = [('A','X'),('W','I'),('V','D'),('U','E'),('T','G'),('S','H'),('R','M'),('Q','C'),('P','L')]

**\*Bombe\_Testing>** let stecker2 = [('T','G'),('C','E'),('K','F'),('D','S'),('R','H'),('O','L'),('N','N'),('B','V'),('A','U'),('P','M'),('Q','Q')]

**\*Bombe\_Testing>** let stecker3 = [('U','U'),('N','N'),('Y','B'),('V','E'),('T','T'),('I','I'),('L','L'),('O','O'),('Z','A'),('H','H'),('C','X')]

**\*Bombe\_Testing>** enigmaEncodeMessage x1 (SteckeredEnigma rotor1 rotor2 rotor3 reflectorB stecker1) (0,0,0)

"AIDEGHMC"

**\*Bombe\_Testing>** enigmaEncodeMessage c2 (SteckeredEnigma rotor1 rotor2 rotor3 reflectorB stecker2) (0,0,5)

"COMPUTERSCJENCESHEFFJELDUNJVERSJTYSTOPTHJSJSTHETESTTWOMESSAGESTOP"

**\*Bombe\_Testing>** enigmaEncodeMessage c3 (SteckeredEnigma rotor1 rotor2 rotor3 reflectorB stecker3) (1,2,3)

"TUZIMGBOMBEHASKELLSIMULATIONSTYPHERESTHEANSDERFXRTESTOHREESZOP"