**Encryption**

cipher(password, encryptedPassword)

characters ← []

shift = LEN(password)

FOR i ← password

append(ord(i) + shift) to characters

ENDFOR

FOR i ← characters

append(chr(i)) to encryptedPassword

append(str(random.randint(1, 9))) to encryptedPassword

ENDFOR

**Decryption**

cipher2(search, password)

characters ← []

characters2 ← []

shift = LEN(search)

FOR i ← search

append(ord(i) - shift) to characters

ENDFOR

FOR i ← characters

append(chr(i)) to characters2

ENDFOR

characters3 ← (“”.join(characters2))

append(characters3) to password

**Validate Password**

validatePassword(password, validity1)

WHILE validity1 = false

IF LEN(password) < 6 or LEN(password) > 18:

ENDWHILE

ENDIF

ELSE IF i NOT IN “a-z”

ENDWHILE

ENDIF

ELSE IF i NOT IN “A-Z”

ENDWHILE

ENDIF

ELSE IF i NOT IN “0-9”

ENDWHILE

ENDIF

ELSE

validity1 ← true

ENDWHILE

ENDWHILE

RETURN password, validity1

**Validate Username**

validateUsername(username, validity2)

records ← logins

usernames ← []

FOR i IN records

append(i[0]) to usernames

ENDFOR

IF username IN usernames

validity2 ← false

ELSE IF LEN(username) < 1:

validity2 ← false

ELSE

validity2 ← true

ENDIF

RETURN username, validity2

**Person Prefers**

personPrefers(N, prefer, person, class1, class2)

FOR i IN range(N)

IF prefer[person][i] = class1

RETURN false

ENDIF

IF prefer[person][i] = class2

RETURN true

ENDIF

ENDFOR

**Stable Marriage Matching Algorithm**

GaleShapley(prefer, matches2)

N ← 3

freeCount ← N

classFree ← [false FOR i IN range(N)]

personClass ← [-1 FOR i IN range(N)]

matches ← []

WHILE freeCount > 0

class1 ← 0

WHILE class1 < N

IF classFree[class1] = false

ENDWHILE

ENDWHILE

class1 += 1

i = 0

WHILE i < N AND classFree[class1] = false

person ← prefer[class1][i]

IF personClass[person - N] = -1

personClass[person - N] ← class1

classFree[class1] ← true

freeCount -= 1

ELSE

class2 ← personClass[person - N]

IF personPrefers[N, prefer, person, class1, class2] = false

personClass[person - N] ← class1

classFree[class1] ← true

classFree[class2] ← false

ENDIF

ENDIF

i += 1

ENDWHILE

ENDWHILE

FOR i IN range(N)

matches3 ← []

append(i + N) to matches3

append(personClass[i]) to matches3

append(matches3) to matches

ENDFOR

GaleShapley(matches, matches2)

**Creating a Preference List**

createPreferenceList(prefer, matches2)

bmiList ← [bmi, user]

bmiList.sort()

bmiList.reverse()

classPreference ← [bmiList[0], bmiList[1], bmiList[2]]

personPreferences ← userPreferences

preferences ← []

FOR i IN classPreference

IF i[0] IN personPreferences

append(classPreference.index(i) + 3) to preferences

ENDIF

ENDFOR

FOR i IN personPreferences

Preference2 = []

IF i = firstClass

append(0) to preference2

ELSE IF i = secondClass

append(1) to preference2

ELSE IF i = thirdClass

append(2) to preference2

ENDIF

append(preference2.index(i)) to preferences

ENDFOR