Since the discovery of the first antibiotic, penicillin, in 1928, countless lives have been saved by the drugs. Despite being an essential part of modern medicine, the use of antibiotics also leads to the evolution of antibiotic-resistant bacteria. For this reason, it has become important to develop the tools to sequence antibiotic peptides. The way that bacteria produce antibiotics is different than the typical mechanisms for protein synthesis and this has created difficulties for the goal of sequencing. First, the peptide sequences of antibiotics like tyrocidines and gramicidins are cyclic rather than linear. Second, these peptides are not produced by the ribosomes. Instead of following the central dogma, antibiotics cyclic peptides are produced by a protein called non-ribosomal peptide synthetase (NRP synthetase) allowing bacteria to fight off enemies even if errors occur in the machinery of the central dogma.

Since the code for these antibiotics can not be found in the genome scientists have looked for alternative methods for sequencing. The fact that they are cyclic also poses a challenge. The mass spectrometer has been a useful tool in the sequencing of peptides. It works by shattering the molecules and weighing the produced fragments in daltons (Da) which are equivalent to the mass of a proton or neutron. Each amino acid has an approximate integer mass based on the number of protons and neutrons are present in the molecules that make it up. After weighing the fragments, the mass spectrometer returns an experimental spectrum with the fragment masses. An ideal spectrum contains the mass of all the possible subpeptides or fragments that can be produced from breaking the peptide. With such a spectrum we can reconstruct a peptide (or list of possible linear peptides) thereby sequencing the peptide. To solve the biological problem of How to sequence an antibiotic? This problem has been converted to a computational one. The cyclopeptide sequencing problem states, “Given an ideal spectrum, find a cyclic peptide whose theoretical spectrum matches the experimental spectrum”. The first step, however, is to be able to create a theoretical spectrum from a cyclic peptide.

Creating a cyclospectrum has two major steps. The first is to create a list of all the possible subpeptides of the string peptide. Step two is to find the integer masses for each subpeptides and add them to the list spectrum with masses ordered from smallest to largest. This list includes 0 as well. A string of n length will produce $n^2 - n$ number of subpeptides. Cyclicpeptide NQEL, for example, has 12 subpeptides: N, Q, E, L, NQ, QE, EL, LN, NQE, QEL, ELN, and LNQ. Because the peptide is cyclic, subpeptides continue from the end of the string to the first positions. The list of subpeptides, which will end up creating a theoretical spectrum of masses, also includes the original peptide.

```
createSpectrum(peptide)
    string <- peptide+peptide
    Subpeptides(0) <- peptide
    n <- |peptide|-1
    while n > 0
        for i <- 0 to |peptide|
            sub <- string(i to i+n)
            add sub to subpeptides
        n <- n-1
```

The theoretical spectrum is created from a list of the masses of all subpeptides, the mass of the original peptide, and 0. Masses of each string in the list subpeptide are found by adding the known integer masses of each amino that make up the subpeptide. The function *getAminoAcids* returns the mass value of each amino acid in the peptide as it is entered into a dictionary. Once all masses are added to the list spectrum, the list is ordered from smallest to largest mass.

```
    spectrum (0) <-0
        for i <- 0 to |subpeptides|
            subpeptideMass<- 0
            for j <- 0 to |subpeptides(i)|
                aminoAcids<- subpeptide(i)(j)
                mass<- getAminoMasses(aminoAcids)
                subpeptideMass<- subpeptideMass+mass
                add subpeptide mass to spectrum
    sort spectrum
    output spectrum
```

We can also find the linear spectrum for a string peptide. In a separate function, the masses of each amino acid in peptide are added to a list. The linear spectrum is compiled by finding the differences between the mass of each prefix.

In order to answer the computational question that is posed, we need to be able to take a spectrum and produce an unknown peptide from the mass information. One such method to sequence a cyclopeptide is to create a blunt force algorithm that creates all possible peptides and test them against the input experimental spectrum. Alternatively, we can design a faster and more efficient branch-and-bound algorithm. This approach begins with a list, Peptides, consisting of an empty peptide. Each iteration, the list of peptides is rewritten with a new set of peptides by adding each amino acid to a peptide that was previously found to be consistent with the experimental spectrum with *Expand*(peptide). If the mass of the peptide is equal to the mass of the tested peptide, ParentMass, and the cyclospectrum of the peptide matches the input spectrum the peptide is added to a list of final peptides. The linear spectrum of the peptide is also checked against the theoretical spectrum. If the peptide does not pass these tests, it is removed from the list, therefore, limiting the number of peptides tested in the next iteration.

```
CyclopeptideSequencing(Spectrum)
    ParentMass <- maximum value in spectrum
    Peptides ← a set containing only the empty peptide
    FinalPeptides ← empty list of strings
    while Peptides is nonempty
        Peptides ← Expand(Peptides)
        for each peptide Peptide in Peptides
            if Mass(Peptide) = ParentMass
                if createSpectrum(Peptide) = Spectrum
                    add Peptide to FinalPeptides
                remove Peptide from Peptides
            else if Peptide is not consistent with Spectrum
                remove Peptide from Peptides
    output FinalPeptides
```

Using this program we can sequence cyclopeptides from the experimental spectrum of masses like that produced by a mass spectrometer. Creating a program helps us reconstruct these peptides is useful for the identification and research of various antibiotics adding to our understanding of their function and synthesis.

Here is a link to the repl that contains my complete code:
https://repl.it/join/krdibnat-carlinwmn

I give permission to post this on the webpage.