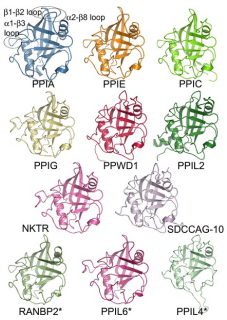
Protein motifs

Proteins perform every practical function in the cell. A structural and functional unit of the protein is a **domain**: in terms of the protein's primary structure, the domain is an interval of amino acids that can evolve and function independently.

Each domain usually corresponds to a single function of the protein (e.g., binding the protein to DNA, creating or breaking specific chemical bonds, …). Some proteins, such as myoglobin and the Cytochrome complex, have only one domain, but many proteins are multifunctional and therefore possess several domains. It is even possible to artificially fuse different domains into a protein molecule with definite properties, creating a **chimeric protein**.

Just like species, proteins can evolve, forming homologous groups called **protein families**. Proteins from one family usually have the same set of domains, performing similar functions.



The human cyclophilin family, as represented by the structures of the isomerase domains of some of its members.

A component of a domain essential for its function is called a **motif**, a term that in general has the same meaning as it does in nucleic acids, although many other terms are also used (blocks, signatures, fingerprints, …). Usually protein motifs are evolutionarily conservative, meaning that they appear without much change in different species.

A part of the domain that is essential for the function is called a **motif**. This term is also used for nucleic acids, although many other terms are often used (blocks, signatures, fingerprints, …). From an evolutionary point of view, protein motifs are usually preserved, which means they exist in multiple forms without all too many differences.

Proteins are identified in different labs around the world and gathered into freely accessible databases. A central repository for protein data is UniProt, which provides detailed protein annotation, including function description, domain structure, and post-translational modifications. UniProt also supports protein similarity search, taxonomy analysis, and literature citations.

Assignment

To allow for the presence of its varying forms, a protein motif is represented by the following shorthand notation. In the shorthand, each uppercase letter represents a specific amino acid. If a series of uppercase letters is enclosed within a pair of square brackets, it corresponds to a single amino acid from the series. The motif `[AC][DEF]G` will thus match the following six protein sequences: `ADG`, `AEG`, `AFG`, `CDG`, `CEG` and `CFG`. If a series of uppercase letters is enclosed within a pair of curly braces, it corresponds to a single amino acid not included in the series. The motif `{AC}` thus represents any amino acid, except `A` or `C`. The lowercase letter `x` is used to represent a single amino acid without any further restrictions.

Using the shorthand notation, a motif is represented as a sequence of groups, where each group belongs to one of four categories as summarized in the table below. We say that a protein sequence matches a given motif, if the number of amino acids of the protein sequence equals the number of groups in the motif, and each amino acid matches with its corresponding group. This way, we see for example that the protein sequence `NFSD` matches the N-glycosylation motif that is written as `N{P}[ST]{P}`.

category	example	matches with
uppercase letter	`A`	the amino acid `A`
lowercase letter `x`	`x`	a single amino acid
series of uppercase letters between square brackets	`[ACD]`	the amino acid `A`, `C` or `D`
series of uppercase letters between curly braces	`{ACD}`	any amino acid, except `A`, `C` or `D`

Your task:

- Write a function `groups` that takes a motif as its string argument. The function must return the number of groups that are contained in the given motif.
- Write a function `match` that takes two string arguments: a protein sequence and a motif. The function must return a Boolean value that indicates whether or not the given protein sequence matches the given motif.
- Write a function `positions` that takes two string arguments: a protein sequence and a motif. The function must return a list containing all positions in the given protein sequence where a match starts with the given motif. These positions must be listed in ascending order.

Example

```
>>> groups('N{P}[ST]{P}')
4
>>> groups('{TCGFSM}{E}[GYD]xSx[YTA]N[AVWMYGCHD]P')
10

>>> match('NFSD', 'N{P}[ST]{P}')
True
>>> match('MFSD', 'N{P}[ST]{P}')
False
>>> match('NPSD', 'N{P}[ST]{P}')
False
>>> match('NFAD', 'N{P}[ST]{P}')
False
>>> match('NFSP', 'N{P}[ST]{P}')
```

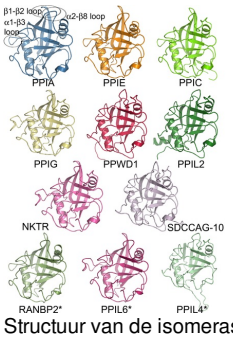
```
False
>>> match('QDNPYIEEIR', '{TCGFSM}{E}[GYD]xSx[YTA]N[AVWMYGCHD]P')
False

>>> positions('MKNKFKTQEELVNHLKTVGFVFANSEIYNGLANAWDYGPLGVLLKNNLKNLWWKEFVTKQKDVVGLDSAIILNPLVWKASGHLDNFSDPLIDCKNCKARYRADKLIESFDENIHIAENSS
[84, 117, 141, 305, 394]
```

Alle praktische functies in een cel worden uitgevoerd door eiwitten. Het zijn echter de **domeinen** die de structurele en functionele eenheden van een eiwit vormen: in termen van de primaire structuur van een eiwit is een domein een interval van aminozuren dat autonoom kan evolueren en functioneren.

Elk domein correspondeert doorgaans met één enkele functie van het eiwit (bv. binden van het eiwit aan DNA, aanmaken of verbreken van specifieke chemische bindingen, …). Sommige eiwitten zoals myoglobine en het cytochroomcomplex hebben slechts één enkel domein, maar de meeste zijn multifunctioneel en beschikken daarvoor over meerdere domeinen. Het is zelfs mogelijk om verschillende domeinen kunstmatig samen te voegen tot een eiwitmolecule met bepaalde eigenschappen. Dit wordt dan een **chimeereiwit** genoemd.

Net zoals soorten kunnen ook eiwitten evolueren, waardoor ze homologe groepen vormen die **eiwitfamilies** genoemd worden. Eiwitten uit dezelfde familie delen meestal dezelfde verzameling domeinen, waardoor ze gelijkaardige functies uitoefenen.



Structuur van de isomerasedomeinen van een aantal eiwitten uit de familie van de humane cyclofilines.

Een onderdeel van een domein dat essentieel is voor de functie wordt een **motief** genoemd. Deze term wordt ook gebruikt bij nucleïnezuren, alhoewel daar ook vaak andere termen gehanteerd worden (blokken, handtekeningen, vingerafdrukken, …). Meestal zijn eiwitmotieven evolutionair gezien zeer geconserveerd, wat betekent dat ze in verschillende soorten voorkomen zonder dat er veel verschillen optreden.

Eiwitten worden geïdentificeerd in laboratoria over de hele wereld en verzameld in vrij toegankelijke databanken. UniProt is één van die centrale opslagplaatsen voor eiwitten, die gedetailleerd informatie beschrijft over hun functies, domeinstructuur en post-translationele modificaties. Aan de hand van UniProt kan je dan bijvoorbeeld zoeken naar gelijkaardige eiwitten, taxonomische analyses uitvoeren of referenties opzoeken in de literatuur.

Opgave

Om alle alternatieve vormen van een eiwitmotief te kunnen voorstellen, wordt het motief genoteerd aan de hand van een verkorte notatie. Hierbij stelt elke hoofdletter een specifiek aminozuur voor. Als een reeks hoofdletters tussen vierkante haakjes staan, dan komt dit overeen met één enkel aminozuur uit de reeks. Het motief `[AC][DEF]G` zal dus matchen met de zes eiwitsequenties: `ADG`, `AEG`, `AFG`, `CDG`, `CEG` en `CFG`. Als een reeks hoofdletters tussen accolades staan, dan komt dit overeen met één enkel aminozuur dat niet in de reeks voorkomt. Het motief `{AC}` kan dus elk aminozuur voorstellen, behalve `A` of `C`. De kleine letter `x` wordt gebruikt voor "een willekeurig aminozuur" dat niet nader gespecifieerd is.

Een motief wordt dus genoteerd als een opeenvolging van groepen, waarbij elke groep behoort tot één van de vier types die hieronder nog eens worden samengevat in tabelvorm. We zeggen dat een gegeven eiwitsequentie matcht met een gegeven motief, als het aantal aminozuren van het eiwit gelijk is aan het aantal groepen in het motief, en elk aminozuur matcht met zijn overeenkomstige groep. Op die manier zien we bijvoorbeeld dat de eiwitsequentie `NFSD` matcht met het N-glycosylatiemotief dat genoteerd wordt als `N{P}[ST]{P}`.

type	**voorbeeld**	**matcht met**
hoofdletter	`A`	het aminozuur `A`
kleine letter `x`	`x`	één enkel aminozuur
reeks hoofdletters tussen vierkante haakjes	`[ACD]`	het aminozuur `A`, `C` of `D`
reeks hoofdletters tussen accolades	`{ACD}`	één enkel aminozuur, behalve `A`, `C` of `D`

Gevraagd wordt:

- Schrijf een functie `groepen` waaraan een motief moet doorgegeven worden. De functie moet teruggeven uit hoeveel groepen het gegeven motief bestaat.
- Schrijf een functie `match` waaraan twee argumenten moeten doorgegeven worden: een eiwitsequentie en een motief. De functie moet een Booleaanse waarde teruggeven die aangeeft of de gegeven eiwitsequentie matcht met het gegeven motief.
- Schrijf een functie `posities` waaraan twee argumenten moeten doorgegeven worden: een eiwitsequentie en een motief. De functie moet een lijst teruggeven van alle posities in de gegeven eiwitsequentie waar er een match gevonden wordt met het gegeven motief. Deze posities wijzen de index in de eiwitsequentie aan waar een match begint, en worden in oplopende volgorde opgelijst.

Voorbeeld

```
>>> groepen('N{P}[ST]{P}')
4
>>> groepen('{TCGFSM}{E}[GYD]xSx[YTA]N[AVWMYGCHD]P')
10

>>> match('NFSD', 'N{P}[ST]{P}')
True
>>> match('MFSD', 'N{P}[ST]{P}')
False
>>> match('NPSD', 'N{P}[ST]{P}')
False
>>> match('NFAD', 'N{P}[ST]{P}')
False
```

```
>>> match('NFSP', 'N{P}[ST]{P}')
False
>>> match('QDNPYIEEIR', '{TCGFSM}{E}[GYD]xSx[YTA]N[AVWMYGCHD]P')
False

>>> posities('MKNKFKTQEELVNHLKTVGFVFANSEIYNGLANAWDYGPLGVLLKNNLKNLWWKEFVTKQKDVVGLDSAIILNPLVWKASGHLDNFSDPLIDCKNCKARYRADKLIESFDENIHIAENSS
[84, 117, 141, 305, 394]
```