

```

/*
Esercizio 1. (punti 3-7) A StrandSet is a component of a model of DNA computer. It consists
of a finite, possibly empty, multi-set of finite, non-empty, sequences on the symbol alph-
abet {A,C,G,T}. The StrandSet is an immutable value with operations that includes:
    Extract. It has two arguments, a Strand p and a StrandSet U, and returns the StrandSet
        that consists of all the Strands of U that contain a sub-sequence equals to p.
        Example: Extract({ACCATAC,TC,TTC,ATCCT},AT) = {ACCATAC, ATCCT}
Use Prolog to define:
a. (points 1) A concrete representation for: Strands and StrandSets
b. (points 4) The predicate extract/3(U,p,V) which holds whenever V=Extract(U,p)
c. (points 1) A concrete representation for Indexed Strands such that each symbol is ind-
    exed by its position in the Strand.
        Example: AlC2C3A4T5A6C7 is the IndexedStrand of ACCATAC
d. (points 2) The predicate indexed/2(p,pi) which holds whenever pi is the representation
    of the Indexed Strand of p.
e. (points 2) The predicate matchI/3(u,p,i) which holds whenever the Strand u has, at the
    position i, one sub-sequence equals to p.
        Example: matchI(ACCATAC, AT,4).
*/

```

-----Answer a.

```

//A Strand is represented by a non-empty list of symbols of the alphabet:
typeStrand([X|Xs):- isAlpha(X), typeStrand1(Xs).
typeStrand1([]).
typeStrand1([X|Xs):- isAlpha(X), typeStrand1(Xs).
//A StrandSet is represented by a, possibly empty, list of Strands
typeStrandSet([]).
typeStrandSet([R|Rs):- typeStrand(R), typeStrandSet(Rs).
//isAlpha/1 is a predicate for the alphabet symbols.
isAlpha(a).
isAlpha(c).
isAlpha(g).
isAlpha(t).

```

-----Answer b.

```

extract(U,P,V):-
    typeStrandSet(U),
    typeStrandSet(V),
    typeStrand(P),
    try(U,P,V).
try([],S,[]).
try([X|Xs],S,[X|Q):-
    match(X,S),
    try(Xs,S,Q).
try([X|Xs],S,Q):-
    \+ match(X,S),
    try(Xs,S,Q).

match([U|Ur],[U|Up):-
    matchA(Ur,Up).
match([_|Ur],P):-
    match(Ur,P).
matchA([_|_],[]).
matchA([U|Ur],[U|Up):-
    matchA(Ur,Up).

```

-----Answer c and d

```

// An Indexed Strand is represented by a non-empty sequence of terms, c(n,u), where n is a
    natural and u is the n-th symbol of the Strand
indexed(U,C):-
    typeStrand(U),

```

```
    index(U,C,1).
index([U|Ur],[c(N,U)|Ir],N):-
    K is N+1,
    index(Ur,Ir,K).
index([],[],_).

-----Answer e
matchI(U,P,I):-
    indexed(U,C),
    matchc(C,P,I).

matchc([c(N,U)|Cr],[U|Up],N):-
    !,
    indexed(Ur,Cr),
    matchA(Ur,Up).
matchc([_|Ur],P,N):- match(Ur,P,N).
```