1. Si definisca una funzione ricorsiva

   $f : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$

   che soddisfi la seguente proprietà

   $(\forall\ x,y \in \mathbb{N}.\ f(x,y) = 2(x+y))$

   in modo che, nella relazione di precedenza indotta dalla definizione data, le catene massimali siano tutte e sole del tipo descritto di seguito:

```
 <x,y>
   |
<x+1,y-1>
   |
<x+2,y-2>
   |
  ...
   |
<x+y,0>
   |
<x+y-1,0>
   |
<x+y-2,0>
   |
  ...
   |
 <0,0>
```

   Dimostrare la correttezza della definizione per induzione ben fondata.

   **Soluzione proposta**

```
let rec f (x, y) = match (x,y) with
     (0,0) -> 0
  |  (n,0) when n>0 -> 2 + f(x-1, y)
  |  (n,m) when m>0 -> f(x+1, y-1)
```

   Dimostrazione di correttezza per induzione ben fondata, utiizzando la relazione di precedenza indotta.

   <u>Caso base</u>

```
  f(0,0)
=   {def. di f, primo pattern}
  0
=   {calcolo}
  2(0+0)
```

Caso induttivo 1

```
  f(n,m)
=   {Ip: n > 0, m=0, def. di f, secondo pattern}
  2 + f(n-1, m)
=   {Ip. induttiva: f(n-1,m) = 2(n-1+m)}
  2 + 2(n-1+m)
=   {calcolo}
  2 + 2(n+m) - 2
=   {calcolo}
  2(n+m)
```

Caso induttivo 2:

```
  f(n,m)
=   {Ip: m>0, def. di f, terzo pattern}
  f(n+1, m-1)
=   {Ip. induttiva: f(n+1, m-1) = 2(n+1+m-1)}
  2(n+1+m-1)
=   {calcolo}
  2(n + m)
```

2. Si definisca una funzione ricorsiva

   `f :  int -> int -> int`

   che soddisfi la seguente proprietà

$$\forall \mathtt{x,y} \in \mathbb{N}.\mathtt{f(x,y) = 2*x+3*y+5}$$

   in modo che, nella relazione di precedenza indotta dalla definizione data, le catene massimali siano tutte e sole del tipo descritto di seguito:

```
<x,y>
  |
<x,y-1>
  |
<x,y-2>
  |
 ...
  |
<x,0>
  |
<x-1,0>
  |
<x-2,0>
  |
 ...
  |
<0,0>
```

Dimostrare la correttezza della definizione per induzione ben fondata.

**Soluzione proposta**

```
    let rec f x y = match (x,y) with
         (0,0) -> 5
      |  (_,m) when m>0 -> 3 + (f x (y-1))
      |  (n,0) when n>0 -> 2 + (f (x-1) y)
```

Dimostrazione di correttezza per induzione ben fondata, utiizzando la relazione di precedenza indotta.

<u>Caso base</u>

```
  f 0 0
=   {def. di f, primo pattern}
  5
=   {calcolo}
  2*0+3*0+5
```

<u>Caso induttivo 1</u>

```
  f n m
=   {Ip: m > 0, def. di f, secondo pattern}
  3 + (f n (m-1))
=   {Ip. induttiva: f n (m-1) = 2*n + 3*(m-1) + 5}
  3 + 2*n + 3*(m-1) + 5
=   {calcolo}
  3 + 2*n + 3*m -3 + 5
=   {calcolo}
  2*n + 3*m + 5
```

<u>Caso induttivo 2:</u>

```
  f n m
=   {Ip: n>0, m=0, def. di f, terzo pattern}
  2 + (f (n-1) m)
=   {Ip. induttiva: f (n-1) m = 2*(n-1) + 3*m + 5}
  2 + 2*(n-1) + 3*m + 5
=   {calcolo}
  2 + 2*n -2 + 3*m + 5
=   {calcolo}
  2*n + 3*m + 5
```

3. Definire una funzione ricorsiva `rep` con tipo

   ```
   rep : 'a -> 'a list -> 'a list -> 'a list
   ```

   in modo che `(rep x xs zs)` sia la lista ottenuta da `xs` rimpiazzando tutte le occorrenze di `x` in `xs` con la sequenza `zs` . Ad esempio:

   ```
   - rep 10 [2;10;12;10;1] [30;40;50] = [2;30;40;50;12;30;40;50;1]
   - rep 'a' [b';'a';'q'] [ ] = ['b';'q']
   - rep 10 [2;1;4;5] [30;40;50] = [2;1;4;5]
   ```

   Dimostrare poi per induzione ben fondata la seguente proprietà

   $(\forall$ `x,xs,zs.` `zs` $\neq$ `[]` $\Rightarrow$ `|(rep x xs zs)|` $\geq$ `|xs|`)

   dove `|xs| indica la lunghezza della lista xs`

   **Soluzione proposta**

   ```
   let rec rep x xs zs = match xs with
         [] -> []
       | y::ys when y<>x -> y :: (rep x ys zs)
       | y::ys when y=x -> zs @ (rep x ys zs)
   ```

Nella dimostrazione per induzione ben fondata della proprietà richiesta, utilizziamo il seguente indebolimento della relazione di precedenza indotta dalla definizione di `rep`:
`(_, xs, _)` $\prec$ `(_, _::xs, _)`
sul dominio delle triple di tipo `'a`$\times$`'a list`$\times$`'a list`. Utilizzeremo inoltre la seguente definizione induttiva:

(i) `|[]| = 0`

(ii) `|x::xs| = 1 + |xs|`

e la seguente (ovvia) proprietà
`|zs@xs| = |zs| + |xs|` (1).

<u>Caso base</u>

```
      |rep x [] zs|
= {def. di rep, primo pattern}
      |[]|
=  {def (i)}
      0
>=
      0
```

Nel seguito, `zs` indicherà una generica lista non vuota e questo ci autorizzerà ad utilizzare, dove necessario, l'ipotesi

$$\texttt{zs} \neq \texttt{[]} \qquad (2).$$

Caso induttivo 1:

```
  |rep x y::ys zs|
=   {Ip: y<>x, def. di rep, secondo pattern}
  |y :: (rep x ys zs)|
=   {def (ii)}
  1 + |(rep x ys zs)|
>=   {(2) e Ip. induttiva: |rep x ys zs| >= |ys|}
  1 + |ys|
=   {def (ii)}
  |y::ys|
```

Caso induttivo 2:

```
  |rep x y::ys zs|
=   {Ip: y=x, def. di rep, terzo pattern}
  |zs @ (rep x ys zs)|
=   {def (1)}
  |zs| + |(rep x ys zs)|
>=   {(2) e Ip. induttiva: |rep x ys zs| >= |ys|}
  |zs| + |ys|
>=   {(2) che implica |zs|>=1}
  1 + |ys|
=   {def (ii)}
  |y::ys|
```