Pensare ricorsivamente
Dalla matematica alla *head-recursion*

Liceo G.B. Brocchi
Classi seconde Scientifico – opzione scienze applicate
Bassano del Grappa, Gennaio 2023

L'*n*-esimo numero naturale, ricorsivamente

- Ecco la funzione che restituisce l'n-esimo numero naturale in modo diretto:

$$natural(n) = n \qquad con\ n\ naturale$$

- Ed ecco un'implementazione ovvia (ed efficiente) in C/C++:

```
unsigned int natural(unsigned int n) {
   return n;
}
```

- Proviamo a vedere la questione in modo ricorsivo (questo ragionamento è solo teorico, ovviamente non si tratta di un calcolo utile in pratica):

l'n-esimo numero naturale è:

0	***se n è uguale a 0***
1 + l'(n - 1)-esimo numero naturale	***altrimenti***

L'*n*-esimo numero naturale, ricorsivamente

```
unsigned long long natural(unsigned long long n) {
  if (n == 0) {
    return 0;
  }
  return 1 + natural(n - 1);
}
```

Per capirla meglio, leggetela così:
***natural(n)* è uguale a 1 + il valore restituito dall'invocazione ricorsiva di *natural* su n – 1.**
Per scrivere funzioni ricorsive dovete prima di tutto *fidarvi* della ricorsione. Poi dimostreremo che non c'è nulla di magico.

```
unsigned long long natural(unsigned int n) {
  return !n ? 0 : 1 + natural(n - 1);
}
```

operatore condizionale ternario
condizione ? istruzioni_se_condizione_vera : istruzioni_se_condizione_falsa

L'*n*-esimo numero naturale: cosa succede sullo stack

la chiamata `natural(0)` porta subito il flusso di controllo della funzione al caso base. Viene restituito il valore 0 al chiamante

return 0

`the caller calls: natural(0)`

L'*n*-esimo numero naturale: cosa succede sullo stack

la chiamata natural(0) porta subito il flusso al caso base. Viene restituito il valore 0 al chiamante

```
the caller gets the value 0
```

L'*n*-esimo numero naturale: cosa succede sullo stack

la chiamata `natural(5)` porta il flusso di controllo della funzione al caso ricorsivo
NB: finora non è stato calcolato niente!

```
the caller calls:
natural(5)
recursive call:
natural(4)
```

return 1 + natural(4)

L'*n*-esimo numero naturale: cosa succede sullo stack

> la chiamata `natural(4)` porta il flusso di controllo della funzione al caso ricorsivo.
> **NB**: finora non è stato calcolato niente!

return 1 + natural(3)

return 1 + natural(4)

```
recursive case:
  natural(3)
```

L'*n*-esimo numero naturale: cosa succede sullo stack

> la chiamata `natural(3)` porta il flusso di controllo della funzione al caso ricorsivo.
> **NB**: finora non è stato calcolato niente!

return 1 + natural(2)

return 1 + natural(3)

return 1 + natural(4)

```
recursive case:
  natural(2)
```

L'*n*-esimo numero naturale: cosa succede sullo stack

> la chiamata `natural(2)` porta il flusso di controllo della funzione al caso ricorsivo.
> **NB**: finora non è stato calcolato niente!

return 1 + natural(1)

return 1 + natural(2)

return 1 + natural(3)

return 1 + natural(4)

```
recursive case:
  natural(1)
```

L'*n*-esimo numero naturale: cosa succede sullo stack

la chiamata `natural(1)` porta di controllo della funzione al caso ricorsivo.
NB: finora non è stato calcolato niente!

return 1 + natural(0)

return 1 + natural(1)

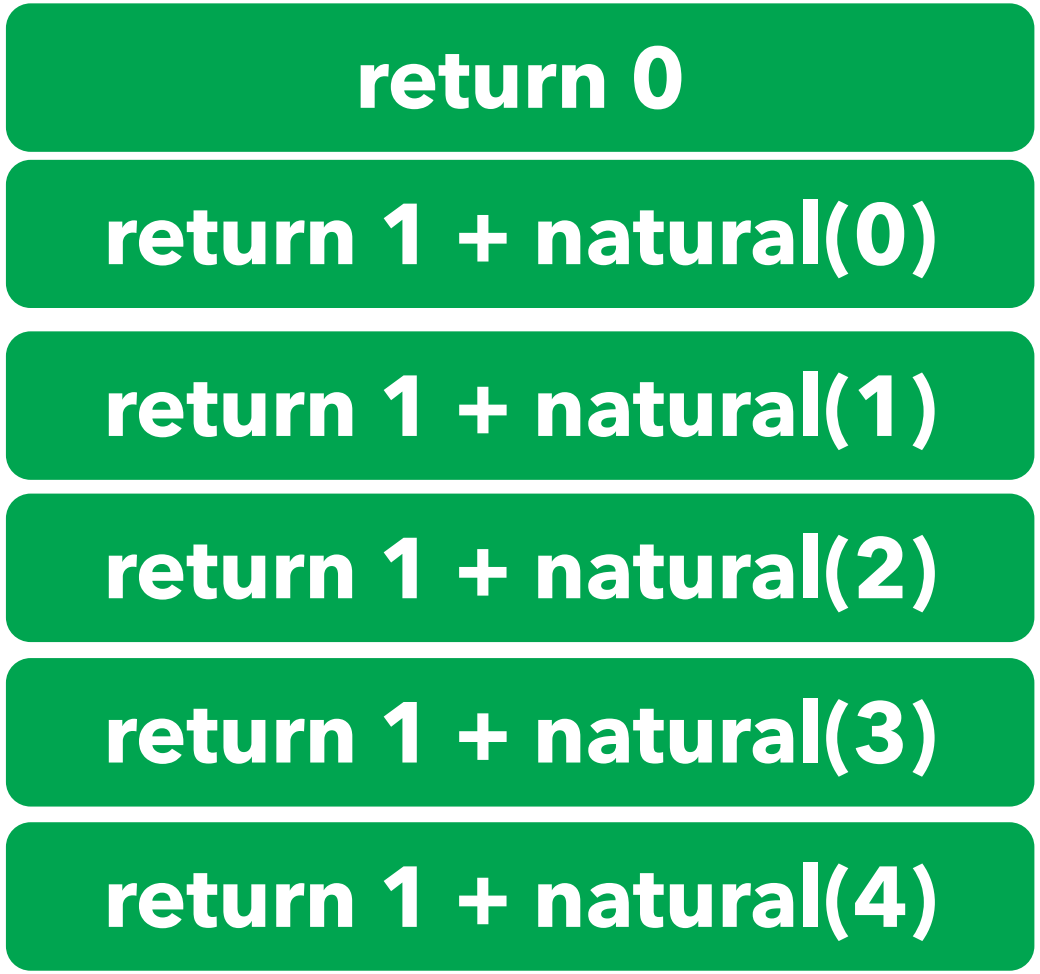
return 1 + natural(2)

return 1 + natural(3)

return 1 + natural(4)

```
recursive case:
  natural(0)
```

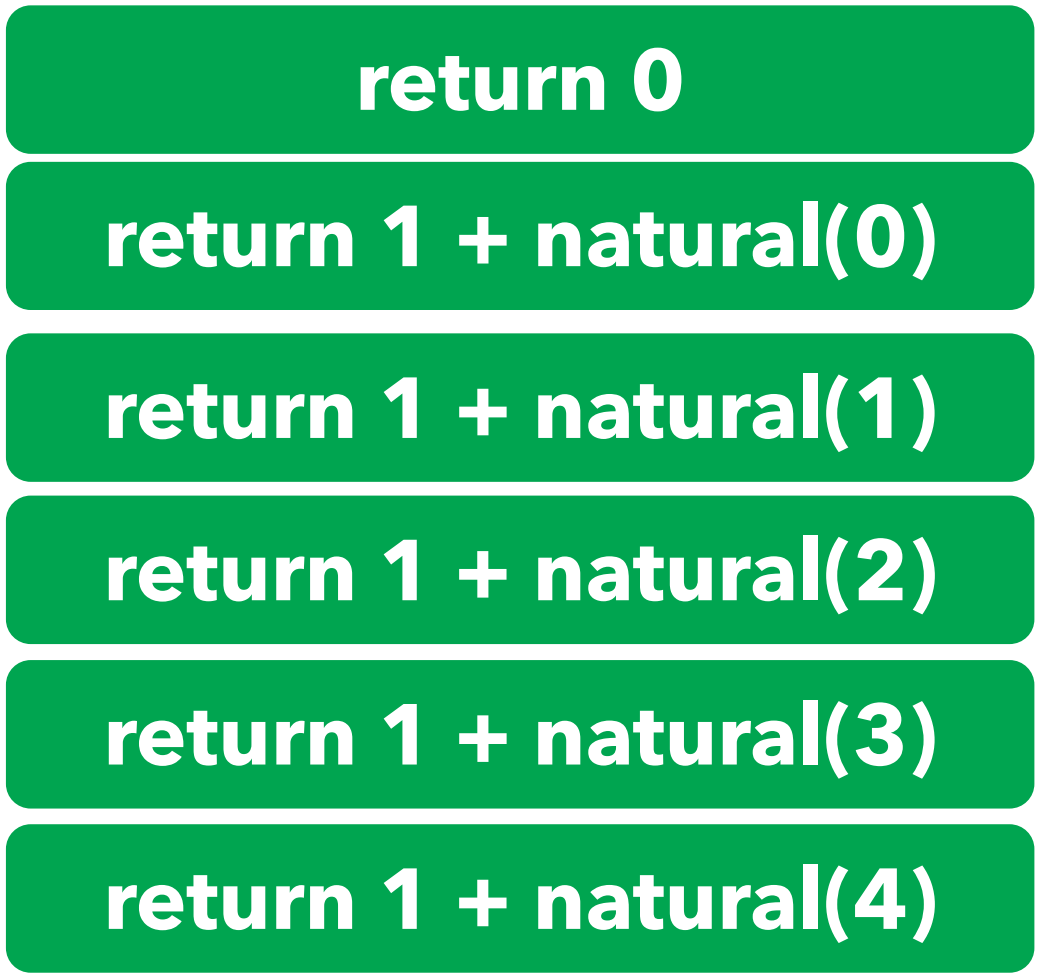
L'*n*-esimo numero naturale: cosa succede sullo stack



la chiamata `natural(0)` porta il flusso al caso base.
NB: finora non è stato calcolato niente!

```
base case: return 0
```

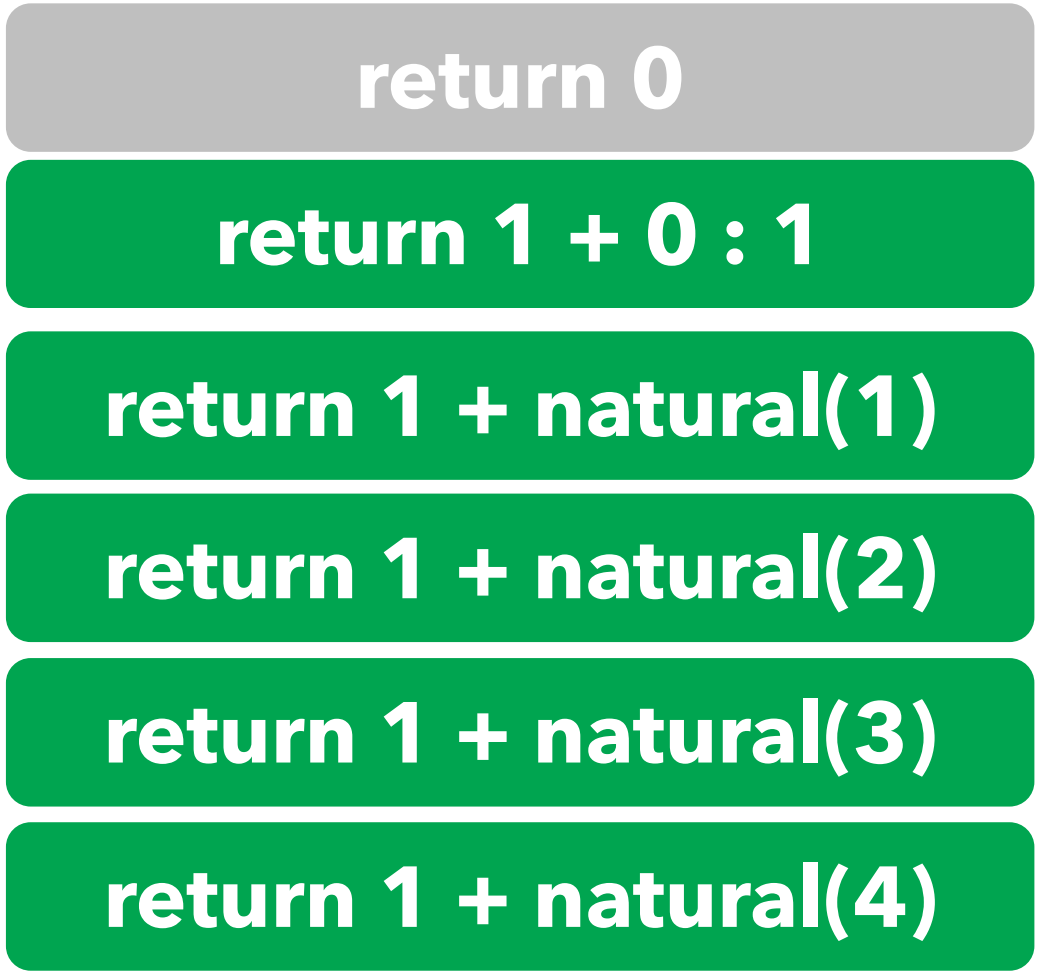
L'*n*-esimo numero naturale: cosa succede sullo stack



una volta raggiunto il caso base, lo stack viene poppato un *frame* alla volta, fino a tornare alla configurazione che aveva al momento della chiamata `natural(5)`. Per semplicità, possiamo pensare che lo stack si «svuoti». I valori restituiti da ogni chiamata vengono utilizzati dal chiamante per effettuare il calcolo.

Il calcolo vero e proprio, di fatto, consiste in: 1 + 1 + 1 + 1 + 1

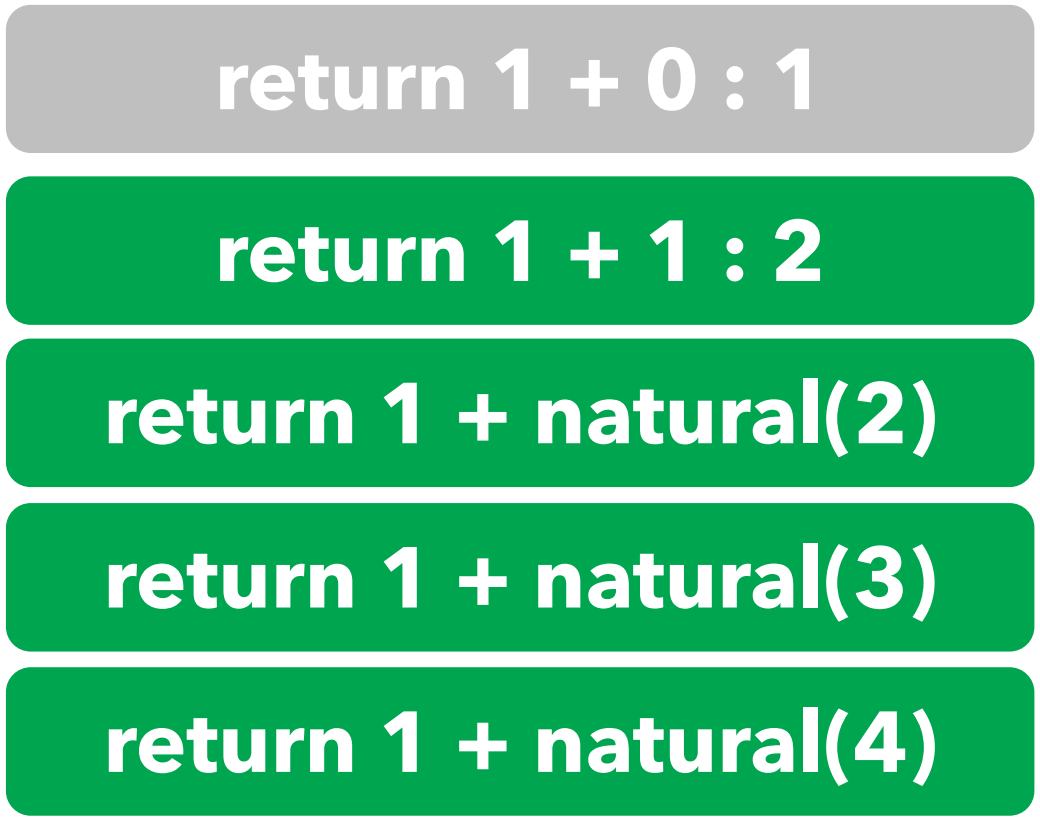
L'*n*-esimo numero naturale: cosa succede sullo stack



una volta raggiunto il caso base, lo stack viene poppato un frame alla volta, fino a tornare alla configurazione che aveva al momento della chiamata `natural(5)`. Per semplicità, possiamo pensare che lo stack si «svuoti». I valori restituiti da ogni chiamata vengono utilizzati dal chiamante per effettuare il calcolo.

Il calcolo vero e proprio, di fatto, consiste in: 1 + 1 + 1 + 1 + 1

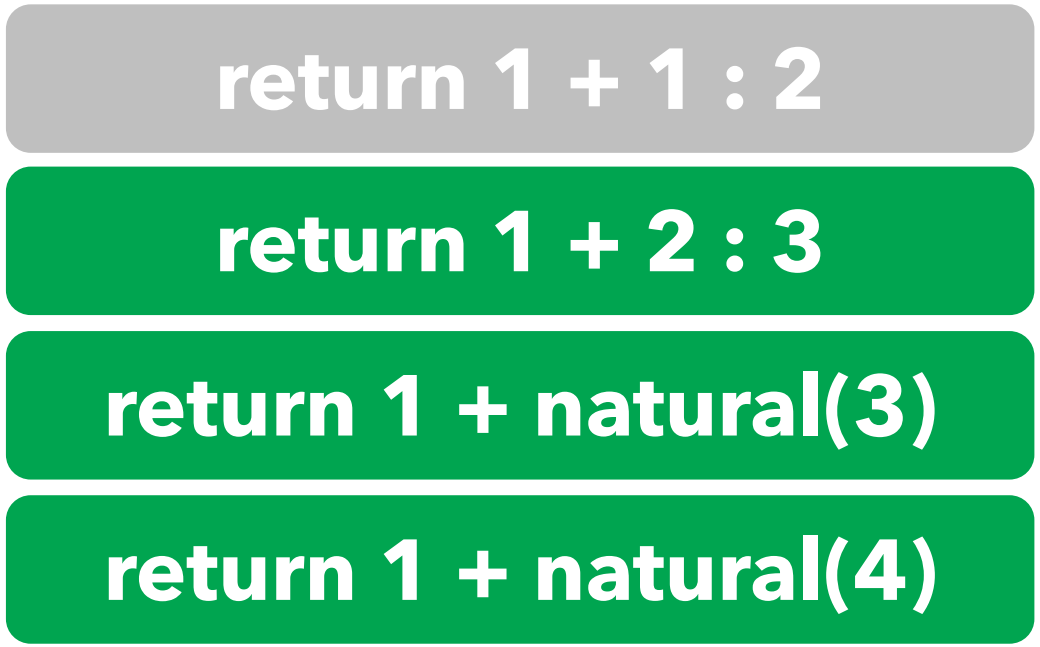
L'*n*-esimo numero naturale: cosa succede sullo stack



una volta raggiunto il caso base, lo stack viene poppato un frame alla volta, fino a tornare alla configurazione che aveva al momento della chiamata `natural(5)`. Per semplicità, possiamo pensare che lo stack si «svuoti». I valori restituiti da ogni chiamata vengono utilizzati dal chiamante per effettuare il calcolo.

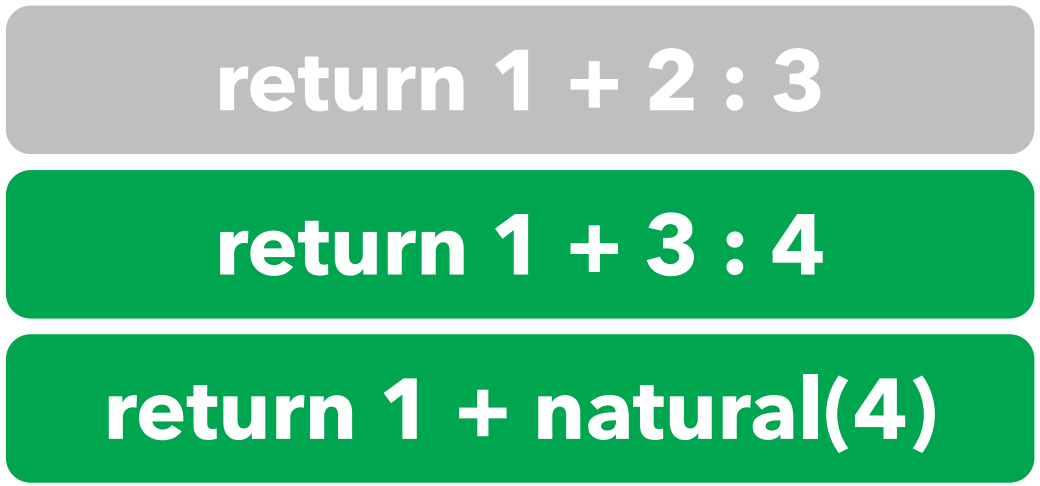
Il calcolo vero e proprio, di fatto, consiste in: 1 + 1 + 1 + 1 + 1

L'*n*-esimo numero naturale: cosa succede sullo stack



una volta raggiunto il caso base, lo stack viene poppato un frame alla volta, fino a tornare alla configurazione che aveva al momento della chiamata `natural(5)`. Per semplicità, possiamo pensare che lo stack si «svuoti». I valori restituiti da ogni chiamata vengono utilizzati dal chiamante per effettuare il calcolo.
Il calcolo vero e proprio, di fatto, consiste in: 1 + 1 + 1 + 1 + 1

L'*n*-esimo numero naturale: cosa succede sullo stack



una volta raggiunto il caso base, lo stack viene poppato un frame alla volta, fino a tornare alla configurazione che aveva al momento della chiamata `natural(5)`. Per semplicità, possiamo pensare che lo stack si «svuoti». I valori restituiti da ogni chiamata vengono utilizzati dal chiamante per effettuare il calcolo.
Il calcolo vero e proprio, di fatto, consiste in: 1 + 1 + 1 + 1 + 1

L'*n*-esimo numero naturale: cosa succede sullo stack

una volta raggiunto il caso base, lo stack viene poppato un frame alla volta, fino a tornare alla configurazione che aveva al momento della chiamata `natural(5)`. Per semplicità, possiamo pensare che lo stack si «svuoti». I valori restituiti da ogni chiamata vengono utilizzati dal chiamante per effettuare il calcolo.
Il calcolo vero e proprio, di fatto, consiste in: 1 + 1 + 1 + 1 + 1

return 1 + 3 : 4

return 1 + 4 : 5

L'*n*-esimo numero naturale: cosa succede sullo stack

il chiamante, che aveva effettuato l'invocazione `natural(5)`, si trova in mano il valore 5, che è proprio quello che voleva.
Il calcolo è stato effettuato in modo «bizzarro» e sicuramente pessimo dal punto di vista dell'efficienza, ma è stato effettuato correttamente

return 5

L’elevamento a potenza con esponente naturale (pseudocodice)

```
pow(b, e):
  if e == 0:
    return 1
  return b * pow(b, e - 1)
```

La moltiplicazione tra naturali (pseudocodice)

```
prod(n, m):
  if m == 0:
    return 0
  return n + prod(n, m - 1)
```

La somma dei primi *n* numeri naturali

```
sum_n(n):
  if n == 0:
    return 0
  return n + sum(n - 1)
```