Esercizi ricorsione

Laboratorio 1

-Fattoriale-

Si scriva una funzione fattoriale(n) che, dato come argomento un intero positivo, restituisca il suo fattoriale.

Esempi

```
fattoriale(1) → 1

fattoriale(5) → 120

fattoriale(10) → 3628800
```

My substring

Scrivere una funzione ricorsiva `mySubstring` che prende come parametri due stringhe `s` e `p` e restituisce true se `p` è in `s`, false altrimenti. Non usare il metodo `substring()`

Esempi:

```
mySubstring(“ciao”,”hello”) -> false

mySubstring(“hello”,”ll”) -> true
```

Palindroma

Scrivere una funzione ricorsiva `palindroma` che prende come parametro una stringa `s` e restituisce true se `s` è palindroma, false altrimenti. Una stringa palindroma si legge nello stesso modo da sinistra a destra che da destra a sinistra.

Esempi:

```
palindroma("ciao") -> false

palindroma("anna") -> true
```

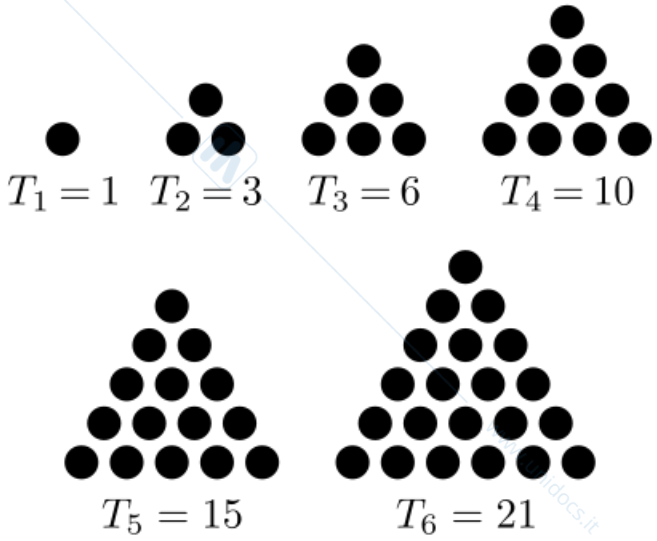
-Numeri triangolari-

Realizzare una funzione ricorsiva che, dato un intero >=0 n, calcoli il valore dell'ennesimo numero triangolare.

Esempi

```
triangle(3) → 6

triangle(6) → 21
```



-Cifre-

Si scriva una funzione ricorsiva cifre(n) che, dato come argomento un intero positivo, mostri a video le sue cifre, una per riga.

Esempi

```
cifre(2563) → 2 5 6 3

cifre(5) → 5

cifre(98763) → 9 8 7 6 3
```

-Cifre in ordine inverso-

Si scriva una funzione ricorsiva cifreInv(n) che, dato come argomento un intero positivo, mostri a video le sue cifre, in ordine inverso, una per riga.

Esempi

```
cifreInv(2563) → 3 6 5 2

cifreInv(5) → 5

cifreInv(98763) → 3 6 7 8 9
```

-Somma cifre-

Si scriva una funzione ricorsiva sommaCifre(n) che, dato come argomento un intero positivo, restituisca la somma delle sue cifre.

Esempi

```
sommaCifre(2563) → 16

sommaCifre(5) → 5

sommaCifre(98763) → 33
```

-Somma array-

Si scriva una funzione ricorsiva sommaArray(a) che, dato come argomento un array di interi, restituisca la somma dei suoi elementi.

Esempi

```
sommaArray([3,6,1,3]) → 13

sommaArray([46,-1,-45, 0,2 -4, 3,-1]) → 0
```

-Sequenza crescente-

Realizzare una funzione ricorsiva che controlli che un array di numeri interi positivi dato in input sia ordinato in ordine strettamente crescente.

Esempi

```
ordinato([1,5,9,12,56,57,29]) → true

ordinato([7]) → true

ordinato([3,7,5,8,9,10]) → false
```

-Fibonacci-

Realizzare una funzione ricorsiva `fib` che, dato un intero positivo `n`, restituisca l'ennesimo numero nella serie di Fibonacci: F(0)=0; F(1)=1; F(n)=F(n-1)+F(n-2), n>=2

Esempi

```
fib(5) → 5

fib(7) → 13
```

Permutazioni

Scrivere una funzione `perm(s)` che prende come parametro una stringa `s` e restituisce un array di stringhe, che contiene tutte le permutazioni dei caratteri in `s` (tutte le stringhe che si possono scrivere con gli stessi caratteri). Potete assumere che i caratteri sono tutti distinti.

Esempio:

```
perm(“abc”) -> [“abc”,”acb”,”bac”,”bca”,”cab”,”cba”]
```

-Piastrelle-

Dobbiamo piastrellare una sentiero largo 1m con delle piastrelle che esistono in 2 dimensioni: 1 × 1 m o 1 × 2 m. Quelle da 1 × 1 possono essere rosse o blu, quelle da 1 × 2 possono essere gialle, verdi o nere. Se il sentiero è lungo n metri, in quanti modi diversi può essere piastrellato? Scrivere una funzione contaPossibilita(n) che restituisca il numero di possibilità, data la lunghezza n. *Suggerimento:* trovare una relazione ricorsiva per calcolare il numero di possibilità, poi implementare la funzione in modo ricorsivo (preferibilmente in coda).

Esempi

```
contaPossibilita(1) → 2

contaPossibilita(7) → 1640
```