

# Esercizi elementari in linguaggio Python

Corso Ottimizzazione Combinatoria (IN440)

Prof. M. Liverani

# Stringhe binarie

Esercizio: Letto in input un numero intero  $n > 0$ , visualizza in output tutte le stringhe binarie con  $n$  cifre

## Suggerimenti:

- $S := (0, 0, \dots, 0)$  può essere implementato con  
`s = [0 for i in range(n)]`

---

## Algoritmo 5 STRINGHEBINARIE( $n$ )

---

**Input:** Un numero intero  $n > 0$

**Output:** Tutte le stringhe binarie  $S := (s_1, s_2, \dots, s_n)$  con  $n$  cifre

1: sia  $S := (0, 0, \dots, 0)$  ( $s_i := 0$  per ogni  $i = 1, 2, \dots, n$ )

2:  $i := 1$

3: **fintanto che**  $i \leq n$  **ripeti**

4:      $i := 1$

5:     **fintanto che**  $i \leq n$  **e**  $s_i \neq 0$  **ripeti**

6:          $s_i := 0$

7:          $i := i + 1$

8:     **fine-ciclo**

9:      $s_i := 1$

10:     scrivi la stringa  $S = (s_1, s_2, \dots, s_n)$

11: **fine-ciclo**

---

# Stringhe binarie

```
n = int(input("Numero di elementi: "))
s = [0 for i in range(n)]
i = 0
while i < n:
    i = 0
    while i < n and s[i] != 0:
        s[i] = 0
        i = i+1
    if i < n: s[i] = 1
print(s)
```



# Permutazioni di un insieme

Esercizio: letto in input  $n > 0$ , visualizza in output tutte le permutazioni degli elementi dell'insieme  $\{1, 2, \dots, n\}$

## Suggerimenti:

per scambiare di posizione gli elementi  $x[i]$  e  $x[j]$  di un array/lista  $x$  è utile definire una funzione:

```
def scambia(x, i, j): ...
```

per ordinare gli elementi  $a[k+1], a[k+2], \dots, a[n]$  dell'array/lista  $a$ :

```
a[k+1:n] = sorted(a[k+1:n])
```

---

## Algoritmo 6 PERMUTAZIONI( $A, n$ )

---

**Input:** L'insieme  $A = \{1, 2, \dots, n\}$

**Output:** Tutte le permutazioni dell'insieme  $A$

```
1: sia  $P := (a_1 = 1, a_2 = 2, \dots, a_n = n)$  la prima permutazione
2: scrivi la permutazione  $P$ 
3: ripeti
4:    $k := n - 1$ 
5:   fintanto che  $k > 0$  e  $a_k > a_{k+1}$  ripeti
6:      $k := k - 1$ 
7:   fine-ciclo
8:   se  $k > 0$  allora
9:      $h := 0$ 
10:    per  $i := k + 1, k + 2, \dots, n$  ripeti
11:      se  $a_i > a_k$  e ( $h = 0$  o  $a_i < a_h$ ) allora
12:         $h := i$ 
13:      fine-condizione
14:    fine-ciclo
15:    scambia  $a_k$  e  $a_h$ 
16:    ordina in ordine crescente gli elementi  $a_{k+1}, a_{k+2}, \dots, a_n$ 
17:    scrivi la nuova permutazione  $P = (a_1, a_2, \dots, a_n)$ 
18:  fine-condizione
19: fintanto che  $k > 0$ 
```

---

# Permutazioni di un insieme

```
def scambia(x, i, j):  
    y = x[i]  
    x[i] = x[j]  
    x[j] = y  
    return
```

```
cont = 1  
n = int(input("Numero di elementi: "))  
a = [i for i in range(1,n+1)]  
print(a, "permutazione n.", cont)  
cont = cont+1  
k = n-2  
while k>=0:  
    k = n-2  
    while k>=0 and a[k]>a[k+1]:  
        k = k-1  
    if k>=0:  
        h = -1  
        for i in range(k+1, n):  
            if a[i]>a[k] and (h==-1 or a[i]<a[h]):  
                h = i  
        scambia(a, k, h)  
        a[k+1:n] = sorted(a[k+1:n])  
        print(a, "permutazione n.", cont)  
        cont = cont+1
```



# Combinazioni di $n$ elementi in classi di $k$

Esercizio: letti in input  $n > 0$  e  $k > 0$  ( $k \leq n$ )  
visualizzare in output tutte le combinazioni in  
classi di  $k$  dell'insieme  $\{1, 2, \dots, n\}$ : ossia tutti  
i sottoinsiemi di  $\{1, 2, \dots, n\}$  con soli  $k$   
elementi

---

## Algoritmo 7 COMBINAZIONI( $A, n, k$ )

---

**Input:** L'insieme  $A = \{a_1, a_2, \dots, a_n\}$  e un intero  $k \leq n$

**Output:** Le combinazioni in classi di  $k$  degli  $n$  elementi di  $A$

```
1: per  $i := 1, 2, \dots, k$  ripeti
2:    $c_i := i$ 
3: fine-ciclo
4: scrivi  $\{a_{c_j}\}_{j:=1, \dots, k}$ 
5:  $i := k$ 
6: fintanto che  $i > 0$  ripeti
7:    $i := k$ 
8:   fintanto che  $i > 0$  e  $c_i = n - (k - i)$  ripeti
9:      $i := i - 1$ 
10:  fine-ciclo
11:  se  $i > 0$  allora
12:     $c_i := c_i + 1$ 
13:    per  $j := i + 1, i + 2, \dots, k$  ripeti
14:       $c_j := c_{j-1} + 1$ 
15:    fine-ciclo
16:    scrivi  $\{a_{c_j}\}_{j:=1, \dots, k}$ 
17:  fine-condizione
18: fine-ciclo
```

---

# Combinazioni di $n$ elementi in classi di $k$

```
n = int(input("Numero di elementi (n): "))
k = int(input("Cardinalità dei sottoinsiemi (k): "))
a = [i for i in range(1,n+1)]
c = [i for i in range(k)]
for i in range(k):
    print(a[c[i]], end=" ")
print()
i = k-1
while i>=0:
    i = k-1
    while i>=0 and c[i] == n-(k-i):
        i = i-1
    if i>=0:
        c[i] = c[i]+1
        for j in range(i+1, k):
            c[j] = c[j-1]+1
        for i in range(k):
            print(a[c[i]], end=" ")
    print()
```

