

# MODULO UTILITA

all'interno del seguente modulo sono presenti in totale 9 funzioni, ovvero:

controllare\_caratteri\_stringa; concatenare\_due\_stringhe; calcolare\_dimensione\_array; controllare\_caratteri\_stringa;  
convertire\_minuscolo\_maiuscolo; convertire\_numeri\_in lettere; calcolare\_radice\_quadrata; calcolare\_resto\_intero;  
convertire\_lettera\_in\_numero

---

## FUNZIONE controllare\_caratteri\_stringa

### INPUT:

- str, stringa di cui controllare i caratteri, stringa
- caratteri\_da\_controllare, caratteri da controllare se presenti in str, array di caratteri

### OUTPUT:

- esito, indica se str contiene almeno uno dei caratteri dati, booleano

### DATI DI LAVORO:

- i, indice per scorrere str, numero naturale > 0
- j, indice per scorrere caratteri\_da\_controllare, numero naturale > 0

### PSEUDOCODICE:

```
esito = VERO
i = 1

MENTRE(i <= calcolare_lunghezza_array_caratteri(caratteri_da_controllare) AND esito <> FALSO)
  j = 1
  MENTRE(j <= stringa_leggere_dimensione(str) AND esito <> FALSO)
    SE(stringa_leggere_carattere(str, j) = i'esima posizione di caratteri_da_controllare OR
stringa_leggere_carattere(str, j) < 32 OR stringa_leggere_carattere(str, j) > 127)
      ALLORA esito = FALSO
    FINE
    j = j + 1
  FINE
  i = i + 1
FINE
```

---

## FUNZIONE controllare\_stringhe\_uguali

### INPUT:

- str1, prima stringa su cui eseguire il controllo, stringa
- str2, seconda stringa su cui eseguire il controllo, stringa

### OUTPUT:

- esito, indica se le due stringhe sono uguali, booleano

### DATI DI LAVORO:

- i, indice per scorrere str, numero naturale > 0

### PSEUDOCODICE:

```
esito = VERO
SE(stringa_leggere_dimensione(str1) = stringa_leggere_dimensione(str2))
  ALLORA
    i = 1
```

```

    MENTRE(i < stringa_leggere_dimensione(str1) AND esito != VERO)
        SE(convertire_minuscolo_maiuscolo(stringa_leggere_carattere(str1, i)) <>
convertire_minuscolo_maiuscolo(stringa_leggere_carattere(str2, i)))
            ALLORA esito = VERO
        FINE
        i = i + 1
    FINE
FINE

```

## FUNZIONE concatenare\_due\_stringhe

### INPUT:

- str1, caratteri da concatenare, array di caratteri
- str2, caratteri da concatenare, array di caratteri

### OUTPUT:

- stringa\_finale, stringa finale nata dalla concatenazione delle prime due, array di caratteri

### DATI DI LAVORO:

- dimensione1, dimensione di str1, numero naturale > 0
- dimensione2, dimensione di str2, numero naturale > 0
- i, indice per scorrere le dimensioni, numero naturale > 0
- dimensione\_stringa\_finale, rappresenta la dimensione della stringa finale, numero naturale > 0

### PSEUDOCODICE:

```

i = 1
j = 1

dimensione1 = calcolare_dimensione_array(str1)
dimensione2 = calcolare_dimensione_array(str2)

dimensione_stringa_finale = dimensione1 + dimensione2

MENTRE(i <= dimensione1)
    i'esimo elemento di stringa_finale = i'esimo elemento di str1
    i = i + 1
FINE

MENTRE(j <= dimensione2)
    (i + j)'esimo elemento di stringa_finale = j'esimo elemento di str2
    j = j + 1
FINE
(i + j)'esimo elemento di stringa_finale = '\0'

```

## FUNZIONE calcolare\_lunghezza\_array\_caratteri

### INPUT:

- caratteri, insieme di caratteri, array di caratteri

### OUTPUT:

- dim, dimensione dell'array di caratteri, numero naturale > 0

### PSEUDOCODICE:

```

dim = 1
MENTRE(dim'esimo elemento di caratteri <> '\0')

```

```
dim = dim + 1  
FINE
```

---

## FUNZIONE convertire\_minuscolo\_maiuscolo

### INPUT:

- lettera, lettera da convertire dal minuscolo al maiuscolo, carattere

### OUTPUT:

- lettera\_convertita, lettera convertita da minuscolo a maiuscolo, carattere

### PSEUDOCODICE:

```
lettera_convertita = lettera  
SE ((lettera_convertita >= CHAR_a) AND (lettera_convertita <= CHAR_z))  
    ALLORA lettera_convertita = lettera - 32  
FINE
```

---

## FUNZIONE convertire\_numeri\_in lettere

### INPUT:

- numero, numero che verrà convertito in lettera, numero naturale >0

### OUTPUT:

- lettera, lettera che rappresenta il numero convertito, carattere

### PSEUDOCODICE:

```
lettera = ' '  
SE(numero >= 10 && numero <= DIM_GRIGLIA_GRANDE)  
    ALLORA lettera = numero + 55  
    ALTRIMENTI  
        SE(numero >= 0 && numero <= 9)  
            ALLORA lettera = numero + 48  
        FINE  
FINE
```

---

## FUNZIONE calcolare\_radice\_quadrata

### INPUT:

- radicando, numero da operare con la radice, numero naturale

### OUTPUT:

- radice, risultato dell' operazione, numero intero

### DATI DI LAVORO:

- radice\_quad, numero che rappresenta la radice quadrata, numero intero
- trovato, valore booleano, booleano

### PSEUDOCODICE:

```
trovato = FALSO  
radice = 2  
radice_quad = radice * radice
```

```
MENTRE(radicando >= radice_quad AND trovato = FALSO)
  SE(radice_quad = radicando)
    ALLORA
      trovato = VERO
      radice = radice - 1
  FINE
  radice = radice + 1
  radice_quad = radice * radice
FINE
SE(trovato = FALSO)
  ALLORA radice = 0
FINE
```

---

## FUNZIONE calcolare\_resto\_intero

### INPUT:

- numeratore, numeratore dell'operazione, numero naturale
- denominatore, denominatore dell'operazione, numero naturale

### OUTPUT:

- risultato, resto della divisione, numero intero

### PSEUDOCODICE:

```
MENTRE(numeratore >= denominatore)
  numeratore = numeratore - denominatore
FINE
```

---

## FUNZIONE convertire\_lettera\_in\_numero

### INPUT:

- lettera, lettera che verrà convertita in numero, carattere

### OUTPUT:

- numero, numero che rappresenta la lettera convertita, numero naturale > 0

### PSEUDOCODICE:

```
SE(lettera >= CHAR_A AND lettera <= CHAR_G)
  ALLORA numero = lettera - 55
ALTRIMENTI
  SE(lettera >= CHAR_0 AND lettera <= CHAR_9)
    ALLORA numero = lettera - 48
  FINE
FINE
```