

Modulo Converter

Descrizione del modulo: Funzioni per conversioni di dati da un tipo all'altro e per utility di vario genere.

INTERFACCIA

| | |
|------------------------|---|
| toUpperCase..... | 1 |
| getLength..... | 1 |
| numberToString | 2 |
| remainder..... | 3 |
| stringToNumber | 3 |
| getIntegerColumn | 4 |
| getCharColumn | 5 |
| concatString | 5 |

toUpperCase

INPUT:

| Nome | Descrizione | Tipo | Vincoli |
|----------------|---|---------------|------------------------|
| letter | Lettera da trasformare | carattere | Compresa tra 'a' e 'z' |
| LOWER_TO_UPPER | Conversione delle parole da minuscolo a maiuscolo o viceversa | Numero intero | 32 |

OUTPUT:

| Nome | Descrizione | Tipo | Vincoli |
|--------|---------------------|-----------|------------------------|
| letter | Lettera trasformata | carattere | Compresa tra 'A' e 'Z' |

ALGORITMO:

SE ((letter >= 'a') AND (letter <= 'z'))

ALLORA

Codice Ascii di letter := Codice Ascii di letter – LOWER_TO_UPPER

FINE

getLength

INPUT:

| Nome | Descrizione | Tipo | Vincoli |
|------|-----------------------------------|---------|---------|
| str | Stringa da calcolare la lunghezza | Stringa | \ |

OUTPUT:

| Nome | Descrizione | Tipo | Vincoli |
|------|-------------------------|--------|------------|
| i | Lunghezza della stringa | intero | $i \geq 0$ |

ALGORITMO:

$i := 0$

MENTRE (elemento in posizione $(i + 1)$ di str \neq ENDSTRING)

$i := i + 1$

FINE

numberToString**INPUT:**

| Nome | Descrizione | Tipo | Vincoli |
|-----------------|------------------------------------|---------------|---------|
| number | Numero da convertire | intero | \ |
| MIN_DIGIT | 0 in codice ASCII | Numero intero | 48 |
| START_ROW_COORD | Inizio della riga nelle coordinate | Numero intero | 3 |

OUTPUT:

| Nome | Descrizione | Tipo | Vincoli |
|--------|--|---------|---------|
| buffer | Stringa che contiene il valore di number | Stringa | \ |

LAVORO:

| Nome | Descrizione | Tipo | Vincoli |
|------|--|---------------|------------|
| i | Posizione della cifra numerica all'interno della stringa | Numero intero | $i > 0$ |
| last | Posizione del carattere di ENDSTRING | intero | $last > 0$ |

ALGORITMO:

$i := 1$

$last := START_ROW_COORD - 1$

SE (number > 9)

ALLORA

$i := \text{START_ROW_COORD} - 1$

$\text{last} := \text{START_ROW_COORD}$

FINE

MENTRE ($\text{number} > 0$)

Codice ASCII dell'elemento in posizione i di buffer $:= \text{remainder}(\text{number}, 10) + \text{MIN_DIGIT}$

$\text{number} := \text{number} / 10$

$i := i - 1$

FINE

elemento in posizione last di buffer $:= \text{ENDSTRING}$

remainder

INPUT:

| Nome | Descrizione | Tipo | Vincoli |
|----------|---------------------------|--------|------------------|
| dividend | Dividendo dell'operazione | intero | \ |
| divider | Divisore dell'operazione | Intero | divider $\neq 0$ |

OUTPUT:

| Nome | Descrizione | Tipo | Vincoli |
|----------|-----------------------|--------|---------|
| dividend | Resto della divisione | intero | \ |

ALGORITMO

MENTRE ($\text{dividend} \geq \text{divider}$)

$\text{dividend} := \text{dividend} - \text{divider}$

FINE

stringToNumber

INPUT:

| Nome | Descrizione | Tipo | Vincoli |
|-----------|---|---------------|---------|
| str | Stringa da convertire in un numero intero | Stringa | \ |
| len | Lunghezza della stringa da convertire | Intero | \ |
| MIN_DIGIT | 0 in codice ASCII | Numero intero | 48 |
| MAX_DIGIT | 9 in codice ASCII | Numero intero | 57 |

OUTPUT:

| Nome | Descrizione | Tipo | Vincoli |
|--------|-------------------|--------|---------|
| number | Numero convertito | intero | \ |

LAVORO:

| Nome | Descrizione | Tipo | Vincoli |
|------|-------------------------------------|--------|---------|
| i | Indice di scorrimento della stringa | intero | $i > 0$ |

ALGORITMO:

i := 1

number := 0

MENTRE (i <= len)

SE ((codice ASCII dell'elemento in posizione i di str >= MIN_DIGIT) AND (codice ASCII dell'elemento in posizione i di str <= MAX_DIGIT))

ALLORA

number := (codice ASCII dell'elemento in posizione i di str – MIN_DIGIT) + number *

10

FINE

i := i + 1

FINE

getIntegerColumn**INPUT:**

| Nome | Descrizione | Tipo | Vincoli |
|-----------------------|--|---------------|----------------------------------|
| column | Carattere da convertire | Carattere | Contiene un valore tra 'A' e 'P' |
| START_UPPERCASE_ASCII | Prima lettera dell'alfabeto maiuscola in ASCII | Numero intero | 65 |

OUTPUT

| Nome | Descrizione | Tipo | Vincoli |
|---------------|-------------------------------------|--------|---|
| numericColumn | Valore di column tradotto in numero | Intero | Column >= TABLE_MIN AND column <= TABLE_MAX |

ALGORITMO:

numericColumn := column – START_UPPERCASE_ASCII + 1

getCharColumn**INPUT:**

| Nome | Descrizione | Tipo | Vincoli |
|-----------------------|--|---------------|---|
| column | Carattere da convertire | Intero | Column >= TABLE_MIN AND column <= TABLE_MAX |
| START_UPPERCASE_ASCII | Prima lettera dell'alfabeto maiuscola in ASCII | Numero intero | 65 |

OUTPUT

| Nome | Descrizione | Tipo | Vincoli |
|------------|--|-----------|------------------------|
| charColumn | Posizione della colonna convertita nel suo carattere associato | Carattere | Compreso tra 'A' e 'P' |

ALGORITMO:

charColumn := column + START_UPPERCASE_ASCII - 1

concatString**INPUT**

| NOME | DESCRIZIONE | TIPO | VINCOLI |
|------|---------------------------------------|---------|---------|
| Str1 | Prima stringa | Stringa | / |
| Str2 | Seconda stringa da concatenare a Str1 | Stringa | / |

OUTPUT

| NOME | DESCRIZIONE | TIPO | VINCOLI |
|------|---------------------------------------|---------|---------|
| Str1 | Prima stringa con seconda concatenata | Stringa | / |

LAVORO

| NOME | DESCRIZIONE | TIPO | VINCOLI |
|------|-----------------------|--------|---------|
| i | Indice di scorrimento | Intero | / |
| J | Indice di scorrimento | Intero | / |

ALGORITMO

Modulo Converter

$I := \text{getLength}(\text{Str1}) + 1$

$J := 1$

MENTRE (j-esimo carattere di STR2 <> ENDSTRING)

 i-esimo carattere di STR1 := j-esimo carattere di STR2

$i := i + 1$

$j := j + 1$

FINE

i-esimo carattere di STR1 := ENDSTRING