

RPG

testare i numeri



Marco Riva

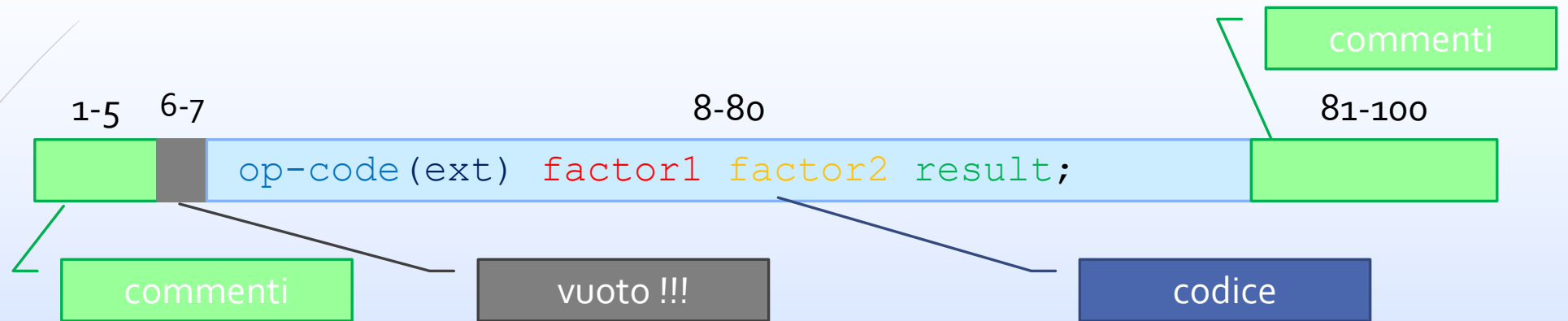


www.markonetools.it



Ultimo aggiornamento: 22/01/2021

Regole base



- ▶ ogni istruzione deve terminare con il carattere **;**
- ▶ `/free ... /end-free` **non** sono più necessari da **7.1 TR7**



Codici operativi "obsoleti"

- ▶ **ATTENZIONE:** non tutti i codici operativi possono essere scritti in formato libero: `ADD`, `ANDxx`, `CABxx`, `CALL`, `CALLB`, `CASxx`, `CAT`, `COMP`, `DEFINE`, `DIV`, `DOUxx`, `DOWxx`, `GOTO`, `IFxx`, `MOVE`, `MOVEA`, `MOVEL`, `MULT`, `MVR`, `ORxx`, `PARM`, `PLIST`, `SCAN`, `SETOFF`, `SETON`, `SQRT`, `SUB`, `SUBST`, `TAG`, `TESTB`, **`TESTN`**, **`TESTZ`**, `TIME`, `WHENxx`, `Z-ADD`, `Z-SUB`
- ▶ I codici operativi non gestiti dal formato libero possono essere definiti "obsoleti" e "deprecabili"



3

come controllare se è un numero valido senza
`TESTN`, `TESTZ`

Testare la validità di un numero

- ▶ come testare se il contenuto di un campo alfanumerico contiene un numero valido?
- ▶ in RPG/400 si usavano i codici operativi TESTN, TESTZ
- ▶ con questi codici operativi i 3 indicatori risultato restituiscono informazioni sul risultato
- ▶ l'indicatore del **maggiore** (pos. 71-72) se ***on** significa che il **numero è valido**



4

TESTN: esempio

Valore	HI	LO	EQ
tutti blanks	*off	*off	*on
00001234	*on	*off	*off
1234	*off	*off	*off
1234,56	*off	*off	*off
1.234,56	*off	*off	*off
-1234g	*off	*off	*off
A0001234	*off	*off	*off
-1234	*off	*off	*off
1234-	*off	*off	*off
1234,56-	*off	*off	*off
-1234,56	*off	*off	*off

- da questi esempi vediamo che in **un solo** caso di quelli elencati TESTN riconosce la stringa come numero valido



come migliorare questo test?



5

Funzione %check

- ▶ con la funzione **%check** è possibile controllare se una stringa contiene caratteri diversi da '0123456789'
- ▶ se il risultato della funzione `%check` è **0** significa che il campo alfanumerico contiene un numero valido
- ▶ la percentuale di successo degli esempi precedenti è equivalente all'uso di TESTN

```
dcl-s f1 char(8);  
dcl-s i uns(3);  
dcl-s elnum char(10) inz('0123456789');  
  
i = %check(elnum:f1);  
  
*inlr = *on;
```



Funzione %dec + monitor

- ▶ Il metodo che ritengo migliore è usare la funzione **%dec** in combinazione con il gruppo **monitor**
- ▶ **%dec** tenta di effettuare la conversione di una stringa di caratteri in un numero
- ▶ il gruppo **monitor** consente di intercettare eventuali errori (status code 105) sulla conversione
- ▶ la percentuale di successo degli esempi precedenti è notevole

Valore	Risultato
tutti blanks	errore
00001234	1234,00
1234	1234,00
1234,56	1234,56
1.234,56	errore
-1234g	errore
A0001234	errore
-1234	-1234,00
1234-	-1234,00
1234,56-	-1234,56
-1234,56	-1234,56



Funzione %dec + monitor: un esempio

```
dcl-s f1 char(8);  
dcl-s wrk packed(8:2);  
dcl-s errnum ind;  
  
errnum = *off;  
monitor;  
  wrk = %dec(f1:8:2);  
  on-error;  
  errnum = *on;  
endmon;  
  
*inlr = *on;
```



Riferimenti



➤ E-mail aziendale: mriva@sirio-is.it



➤ Blog: www.markonetools.it



➤ E-mail blog: info@markonetools.it



➤ LinkedIn: www.linkedin.com/in/marcoriva-mk1



➤ Twitter: [@MarcoRiva73](https://twitter.com/MarcoRiva73)



➤ Facebook: <https://www.facebook.com/markonetools/>



➤ YouTube: <https://www.youtube.com/channel/UCb47YJOJCzU-5x4nnGzDu-w>

Power coffee - MK1



9