

# The `itnumpar` package

## Il pacchetto `itnumpar`\*

Enrico Gregorio  
Enrico dot Gregorio (at) univr dot it

2007/04/03

### Prima sezione: Introduzione in inglese

Sorry, this is a package intended for people writing in Italian. The title of this section is ‘Introduction in English’. However, I’ll give the flavor of it, if not the whole story.

The name of the package is `itnumpar`: `it` for Italian, `num` for ‘numero’ (number), `par` for ‘parola’ (word).

Its purpose is to give a representation of a number in words. The following commands take as an argument a counter’s name (they are similar to `\arabic`, `\roman` and similar ones:

- `\numeroinparole` for representing the counter’s value in Italian words, with lowercase initial;
- `\Numeroinparole` for representing the counter’s value in Italian words, with uppercase initial;
- `\ordinalem` for representing the counter’s value as a (spelled out) ordinal number in masculine form with lowercase initial;
- `\ordinalef` for representing the counter’s value as a (spelled out) ordinal number in feminine form with lowercase initial;
- `\Ordinalem` for representing the counter’s value as a (spelled out) ordinal number in masculine form with uppercase initial;
- `\Ordinalef` for representing the counter’s value as a (spelled out) ordinal number in feminine form with uppercase initial.

The Italian word ‘ordinale’ means ‘ordinal’, of course; ‘numero in parole’ means ‘number in words’.

The other defined commands are

---

\*This document corresponds to `itnumpar` v1.0, dated 2007/04/03.

<code>\printnumeroinparole</code>	<code>\printNumeroinparole</code>
<code>\printordinalem</code>	<code>\printordinalef</code>
<code>\printOrdinalem</code>	<code>\printOrdinalef</code>

which take a number as an argument and print its textual representation like the commands above.

For example, the present document has been written saying

```
\renewcommand{\thesection}{\Ordinalef{section} sezione:}
```

thus getting the spelled out feminine ordinal in front of the section title (section in Italian is ‘sezione’ and is feminine).

The package (in the present version) supports positive numbers up to 9999 for `\numeroinparole` and `\Numeroinparole`; up to 999 for the ordinal forms. For `\numeroinparole`, zero is admissible. For all the commands, negative numbers give an empty result.

All these commands are fully expandable and their result is a string of character tokens. For example, `\printnumeroinparole{12}` expands to the string of characters ‘dodici’.

## Seconda sezione: Introduzione in italiano

Il nome del pacchetto è `itnumpar` (numeri in parole in italiano). Ci sono altri pacchetti su CTAN che fanno un servizio simile per altre lingue, ma nessuno è facilmente adattabile all’italiano. O forse sì, una volta ho provato ad adattarne uno, ma senza grandi risultati.

Quali sono i problemi? Che alcune vocali si assimilano: “venti–ventuno”, “trentasette–trentotto”, “centoventuno”, “centottantasei”. Che i numeri con cifra delle unità 3 in lettere vogliono l’accento: “ventitré”; con l’unica eccezione di “tre”.

Il pacchetto tratta solo numeri relativamente piccoli; non potrete usarlo per compilare assegni di importo superiore a novemilanovecentonovantanove euro<sup>1</sup>. Tuttavia una famosa opera letteraria in italiano è divisa in cento parti e quindi ho deciso di andare oltre il limite di novantanove di un primo abbozzo apparso sul Forum del G<sub>IT</sub><sup>2</sup>.

Rispetto a quella versione ho introdotto anche la possibilità di scrivere gli ordinali (fino al novecentonovantanovesimo), sia come aggettivi maschili che femminili, come vedete nella numerazione delle sezioni di questo stesso documento.

I comandi che descrivo nella prossima sezione sono completamente espandibili e danno come risultato una lista di *token* fatta di caratteri. Per esempio, l’espansione di `\printnumeroinparole{12}` è la sequenza di caratteri ‘dodici’.

---

<sup>1</sup>In alcuni paesi, non in Italia, è possibile stamparsi i propri assegni.

<sup>2</sup>Gruppo Utenti Italiani di T<sub>E</sub>X, [www.guit.sssup.it](http://www.guit.sssup.it)

## Terza sezione: I comandi

`\numeroinparole`  
`\Numeroinparole`  
`\ordinalem`  
`\ordinalef`  
`\Ordinalem`  
`\Ordinalef`

In analogia con i comandi del tipo `\arabic`, `\Roman` e simili, possiamo rappresentare il valore di un contatore con il numero scritto in parole o con l'ordinale corrispondente, maschile o femminile, con iniziale minuscola o maiuscola; ciascuno prende come argomento il nome di un contatore:

- `\numeroinparole` rappresenta in parole il valore del contatore, con iniziale minuscola;
- `\Numeroinparole` rappresenta in parole il valore del contatore, con iniziale maiuscola;
- `\ordinalem` rappresenta il valore del contatore come numero ordinale maschile con iniziale minuscola;
- `\ordinalef` rappresenta il valore del contatore come numero ordinale femminile con iniziale minuscola;
- `\Ordinalem` rappresenta il valore del contatore come numero ordinale maschile con iniziale maiuscola;
- `\Ordinalef` rappresenta il valore del contatore come numero ordinale femminile con iniziale maiuscola.

Per esempio, nel preambolo di questo documento compare il comando

```
\renewcommand{\thesection}{\Ordinalef{section} sezione:}
```

che spiega il curioso formato dei titoli di sezione.

`\printnumeroinparole`  
`\printNumeroinparole`  
`\printordinalem`  
`\printordinalef`  
`\printOrdinalem`  
`\printOrdinalef`

Ci sono altri comandi a disposizione, che permettono di scrivere in lettere o come ordinale un numero dato come argomento:

```
\printnumeroinparole \printNumeroinparole  
\printordinalem      \printordinalef  
\printOrdinalem      \printOrdinalef
```

Per esempio, con `\printOrdinalef{123}` si ottiene “Centoventitreesima”, mentre con `\printnumeroinparole{6783}` si ottiene “seimilasettecentottantatré”.

Non è possibile ottenere, con questi comandi, il modo antiquato di esprimere gli ordinali: “decimoprimo” invece di “undicesimo”. Per precisa scelta stilistica, si ottiene “centoundicesimo” e non “centundicesimo” come alcuni scrivono<sup>3</sup>. Ovviamente, per stampare il valore di un contatore in parole, si deve scrivere, per esempio

```
\printnumeroinparole{\value{section}}
```

che, dato qui, produce “tre”.

---

<sup>3</sup>Se non vi va, scrivete voi il vostro pacchetto.

## Quarta sezione: Limitazioni

Il comando `\numeroinparole` (e il corrispondente `\printnumeroinparole`) non possono essere usati solo con numeri interi  $m$  tali che  $0 \leq m \leq 9999$ . I comandi per i numeri ordinali funzionano solo con interi  $n$  tali che  $1 \leq n \leq 999$ .

Usati con numeri negativi (e anche zero per le forme ordinali), danno un risultato vuoto, senza messaggio di errore.

Forse future versioni del pacchetto ampliaranno l'intervallo di applicabilità.

## Quinta sezione: Esempi

Esiste, e l'ho aggiunto solo per questa applicazione, una variante `*` del comando `\printnumeroinparole`; i versi seguenti sono stato scritti come

```
\begin{verse}
In Italia \printnumeroinparole{640},\
In Almagna \printnumeroinparole*{231},\
\printNumeroinparole{100} in Francia,
  in Turchia \printnumeroinparole*{91},\
Ma in Ispagna son gi'a \printnumeroinparole{1003}.
\end{verse}
```

ed ecco le immortali parole dell'abate Da Ponte:

```
In Italia seicentoquaranta,
In Almagna duecentotrentuna,
Cento in Francia, in Turchia novantuna,
Ma in Ispagna son già milletré.
```

Supponiamo di voler scrivere con la classe `book`, in modo che i capitoli siano numerati in parole: “Capitolo uno”, “Capitolo due” e così via, ma che poi vogliamo riferirci ai capitoli per numero durante il testo. Dare

```
\renewcommand{\thechapter}{\numeroinparole{chapter}}
```

sarebbe sbagliato, perché i riferimenti sarebbero con la forma testuale e non numerica. Occorre dunque modificare il comando `\chapter` o, per essere precisi, `\@chapter`, che è quello chiamato da `\chapter` se non si usa la variante `*`. Per essere ancora più precisi, occorre modificare `\@makechapterhead`, che è il comando responsabile della stampa del titolo. Invece di modificare del tutto il comando, gli aggiungiamo qualcosa.

```
\makeatletter
\let\orig@makechapterhead\@makechapterhead
\def\@makechapterhead#1{\begingroup
  \def\thechapter{\numeroinparole{chapter}}%
  \orig@makechapterhead{#1}\endgroup}
\makeatother
```

e otteniamo il risultato voluto. L'uso del pacchetto `titlesec` è indicato per ottenere effetti diversi senza troppa fatica. Nel caso della classe `amsbook` occorre un trucco diverso:

```

\makeatletter
\let\orig@makechapterhead\makechapterhead
\def\@makechapterhead#1{\begingroup
  \edef\thechapter{\uppercase{\numeroinparole{chapter}}}%
  \orig@makechapterhead{#1}\endgroup}
\makeatother

```

perché la scritta “capitolo uno” deve essere tutta in maiuscolo.

In entrambe le modifiche, si usa il fatto che `\@makechapterhead` chiama al suo interno `\thechapter` per stampare il numero; lo usiamo allo stesso modo, ma ridefinendo localmente `\thechapter`. Con `amsbook` possiamo usare il fatto che `\numeroinparole` è completamente espandibile, mentre `\uppercase` non lo è.

## Sesta sezione: Implementation

I'll use English to describe the implementation of the macros.

Usual stuff for representing counters `\langle name \rangle` is the user interface, `\@⟨name⟩` is the internal command The capital letter is for a capital initial, ‘m’ is for ‘masculine’, ‘f’ is for ‘feminine’.

```

\numeroinparole
\Numeroinparole 1 ⟨*package⟩
  \ordinalem    2 \def\numeroinparole#1{\expandafter\@numeroinparole\csname c@#1\endcsname}
  \ordinalef    3 \def\Numeroinparole#1{\expandafter\@Numeroinparole\csname c@#1\endcsname}
  \Ordinalem    4 \def\ordinalem#1{\expandafter\@ordinalem\csname c@#1\endcsname}
  \Ordinalef    5 \def\ordinalef#1{\expandafter\@ordinalef\csname c@#1\endcsname}
  \Ordinalem    6 \def\Ordinalem#1{\expandafter\@Ordinalem\csname c@#1\endcsname}
  \Ordinalef    7 \def\Ordinalef#1{\expandafter\@Ordinalef\csname c@#1\endcsname}

\printnumeroinparole \print⟨name⟩ gets as an argument a ⟨number⟩ and produces it in the literal form;
\printNumeroinparole instead of doing six similar definitions, we do it only once
  \printordinalem    8 \begingroup
  \printordinalef    9 \def\np@doprintdef#1{%
  \printOrdinalem   10 \expandafter\xdef\csname print#1\endcsname##1{%
  \printOrdinalef   11   {\count@=#1\relax\csname @#1\endcsname{\count@}}}}
  12 \np@doprintdef{numeroinparole}
  13 \np@doprintdef{Numeroinparole}
  14 \np@doprintdef{ordinalem}
  15 \np@doprintdef{ordinalef}
  16 \np@doprintdef{Ordinalem}
  17 \np@doprintdef{Ordinalef}
  18 \endgroup

```

Just for an application, I define a \* version of `\printnumeroinparole`

```

19 \def\np@oa{o}
20 \let\@printnumeroinparole\printnumeroinparole
21 \def\printnumeroinparole{%
22   \@ifstar{\@printnumeroinparolef}{\@printnumeroinparole}%
23 }
24 \def\@printnumeroinparolef#1{\def\np@oa{a}\printnumeroinparole{#1}}
```

`\@numeroinparole` `\@numeroinparole` must keep care of two particularities: ‘zero’ and ‘tre’, since all numbers ending in ‘3’ must have an accent in the word, except for ‘3’ itself. So we look for the particular cases and then call `\np@numeroinparole` with an empty second argument. `\@Numeroinparole` works in the same way.

```

25 \def\@numeroinparole#1{%
26   \ifnum#1=0 zero% Exception
27   \else
28   \ifnum#1=3 tre% Exception
29   \else
30   \np@numeroinparole{#1}{}%
31   \fi\fi}
32 \def\@Numeroinparole#1{%
33   \ifnum#1=0 Zero% Exception
34   \else
35   \ifnum#1=3 Tre% Exception
36   \else
37   \np@Numeroinparole{#1}{}%
38   \fi\fi}
```

`\np@numeroinparole` `\np@Numeroinparole` `\np@numeroinparole` is defined with two arguments; the first one is the number, the second one is either empty or ‘o’. We support only numbers less than 10000. If  $999 < \# < 10000$ , we call `\np@thousand`; if  $99 < \# < 1000$  we call `\np@hundred`; if  $29 < \# < 100$ , we call `\np@large`; if  $0 < \# < 30$  we call `\np@small`. Numbers from 1 to 19 are special; numbers from 20 to 29 have ‘i’ in the root, numbers from 30 to 99 have ‘a’ in the root: ‘venti’, ‘trenta’, ‘quaranta’ and so on. The same for `\np@Numeroinparole`.

```

39 \def\np@numeroinparole#1#2{%
40   \ifnum#1>9999
41     \ctrerr
42   \else
43     \ifnum#1>999
44       \expandafter\np@thousand\number#1 \np@del
45     \else
46       \ifnum#1>99
47         \expandafter\np@hundred\number#1 \np@del
48       \else
49         \ifnum#1<30
50           \np@small{#1}{#2}%
51         \else
52           \expandafter\np@large\number#1{#2}%
53         \fi\fi\fi\fi}
```

```

54 \def\np@Numeroinparole#1#2{%
55   \ifnum#1>9999
56     \ctrerr
57   \else
58     \ifnum#1>999
59       \expandafter\np@Thousand\number#1 \np@del
60     \else
61       \ifnum#1>99
62         \expandafter\np@Hundred\number#1 \np@del
63       \else
64         \ifnum#1<30
65           \np@Small{#1}{#2}%
66         \else
67           \expandafter\np@Large\number#1{#2}%
68         \fi\fi\fi}

```

`\np@small` `\np@small` is just an `\ifcase` routine; we add in front the second argument, which `\np@Small` is empty if  $1 < \# < 100$  or if the third digit is 0. The same for `\np@Small`, but argument #2 will always be empty.

```

69 \def\np@small#1#2{#2\ifcase#1 \or
70   un\np@oa\or due\or tr\’e\or quattro\or cinque\or
71   sei\or sette\or otto\or nove\or dieci\or
72   undici\or dodici\or tredici\or quattordici\or quindici\or
73   sedici\or diciassette\or diciotto\or diciannove\or venti\or
74   ventuno\or ventidue\or ventitr\’e\or ventiquattro\or venticinque\or
75   ventisei\or ventisette\or ventotto\or ventinove\fi}
76 \def\np@Small#1#2{\ifcase#1 \or
77   Uno\or Due\or\or Quattro\or Cinque\or Sei\or Sette\or
78   Otto\or Nove\or Dieci\or Undici\or Dodici\or Tredici\or
79   Quattordici\or Quindici\or Sedici\or Diciassette\or Diciotto\or
80   Diciannove\or Venti\or Ventuno\or Ventidue\or Ventitr\’e\or
81   Ventiquattro\or Venticinque\or Ventisei\or Ventisette\or
82   Ventotto\or Ventinove\fi}

```

`\np@large` `\np@large` has three arguments: the two digits of # and the second argument to `\np@numeroinparole`. We do a pair of `\ifcase`; notice that the third argument is not put in front of ‘ottant’, since we say ‘centottanta’, not doubling the ‘o’. Since we say ‘trentuno’ and ‘trentadue’, we put the vowel, when needed, in the second part. In a similar way we define `\np@Large`.

```

83 \def\np@large#1#2#3{%
84   \ifcase#1 \or\or\or #3trent\or #3quarant\or #3cinquant\or
85     #3sessant\or #3settant\or ottant\or #3novant\fi
86   \ifcase#2 a\or un\np@oa\or adue\or atr\’e\or aquattro\or
87     acinque\or asei\or asette\or otto\or anove\fi}
88 \def\np@Large#1#2#3{%
89   \ifcase#1 \or\or\or Trent\or Quarant\or Cinquant\or
90     Sessant\or Settant\or Ottant\or Novant\fi
91   \ifcase#2 a\or uno\or adue\or atr\’e\or aquattro\or
92     acinque\or asei\or asette\or otto\or anove\fi}

```

`\np@hundred` `\np@hundred` has two arguments, the third digit and the other two: 123 is split into ‘1’ and ‘23’. We add the digit name in front of ‘cent’ if the digit is  $> 1$ ; if the second argument is zero, we add a final ‘o’ and we are done (cento, duecento, and so on), otherwise we call recursively `\np@numeroinparole` with ‘o’ as its second argument.

```

93 \def\np@hundred#1#2\np@del{%
94   \ifcase#1 \or\or due\or tre\or quattro\or cinque\or
95     sei\or sette\or otto\or nove\fi cent%
96   \ifnum#2=0 o\else\np@numeroinparole{#2}{o}\fi}
97 \def\np@Hundred#1#2\np@del{%
98   \ifcase#1 \or Cent\or Duecent\or Trecent\or Quattrocent\or
99     Cinquecent\or Seicent\or Settecent\or Ottocent\or Novecent\fi
100  \ifnum#2=0 o\else\np@numeroinparole{#2}{o}\fi}

```

`\np@thousand` `\np@thousand` does similarly (1234 is split into ‘1’ and ‘234’); if the fourth digit is 1, we output ‘mille’, otherwise (*digitname*)mila. Next we call recursively `\np@numeroinparole` on the second part.

```

101 \def\np@thousand#1#2\np@del{%
102   \ifnum#1=1 mille\else
103   \ifcase#1 \or\or due\or tre\or quattro\or cinque\or
104     sei\or sette\or otto\or nove\fi mila\fi
105   \ifnum#2=0 \else\np@numeroinparole{#2}{}\fi}
106 \def\np@Thousand#1#2\np@del{%
107   \ifnum#1=1
108     Mille%
109   \else
110   \ifcase#1 \or\or Due\or Tre\or Quattro\or Cinque\or
111     Sei\or Sette\or Otto\or Nove\fi
112     mila\fi
113   \ifnum#2=0 \else\np@numeroinparole{#2}{}\fi}

```

`\@ordinalem` Now we come to the ordinals; each one of the commands calls the generic one `\@ordinalef` adding ‘o’ for the masculine form and ‘a’ for the feminine form.

```

114 \def\@ordinalem#1{\np@ordinal{#1}{o}}
115 \def\@ordinalef#1{\np@ordinal{#1}{a}}
116 \def\@Ordinalem#1{\np@Ordinal{#1}{o}}
117 \def\@Ordinalef#1{\np@Ordinal{#1}{a}}

```

`\np@ordinal` `\np@ordinal` has two arguments, for reasons similar to the above. Ordinals for  $0 < \# < 11$  are very special, they cannot be used in the recursion; so we check for that case and otherwise we call `\np@ordinall`.

```

118 \def\np@ordinal#1#2{%
119   \ifnum#1=0
120     \expandafter\@gobble
121   \else
122     \ifnum#1<11
123     \ifcase#1 \or prim\or second\or terz\or quart\or quint\or
124       sest\or settim\or ottav\or non\or decim\fi

```



```

125 \else
126 \np@ordinall{#1}{#2}%
127 \fi\fi}

The definition of \np@ordinall is similar to that of \np@numeroinparole, but
we support only numbers less than 1000.

128 \def\np@ordinall#1#2{%
129 \ifnum#1>999
130 \ctrerr
131 \else
132 \ifnum#1>99
133 \expandafter\np@hundredord\number#1 \np@del
134 \else
135 \ifnum#1<30
136 \np@smallord{#1}{#2}%
137 \else
138 \expandafter\np@largeord\number#1{#2}%
139 \fi\fi\fi}
140 \def\np@Ordinal#1#2{%
141 \ifnum#1=0
142 \expandafter@gobble
143 \else
144 \ifnum#1<11
145 \ifcase#1 \or Prim\or Second\or Terz\or Quart\or Quint\or
146 Sest\or Settim\or Ottav\or Non\or Decim\fi
147 \else\np@Ordinall{#1}{#2}\fi\fi}
148 \def\np@Ordinall#1#2{%
149 \ifnum#1>999
150 \ctrerr
151 \else
152 \ifnum#1>99
153 \expandafter\np@hundredOrd\number#1 \np@del
154 \else
155 \ifnum#1<30
156 \np@smallOrd{#1}{#2}%
157 \else
158 \expandafter\np@largeOrd\number#1{#2}%
159 \fi\fi\fi}

```

`\np@smallord` `\np@smallord` is similar to `\np@small` (the first cases are used only when called recursively. It produces, e.g., ‘tredecim’, the final letter is added by `\@ordinalm` or `\@ordinalf` or the capitalized forms.

```

160 \def\np@smallord#1#2{%
161 \ifcase#1 \or un\or #2du\or #2tre\or #2quattr\or #2cinqu\or
162 #2sei\or #2sett\or ott\or #2nov\or \or #2undic\or
163 #2dodic\or #2tredic\or #2quattordic\or #2quindic\or
164 #2sedic\or #2diciasset\or #2diciott\or #2diciannov\or
165 #2vent\or #2ventun\or #2ventidu\or #2ventitre\or
166 #2ventiquattr\or #2venticinqu\or #2ventisei\or
167 #2ventisett\or #2ventott\or #2ventinov%

```

```

168 \fi esim}

\np@largeord \np@largeord is similar to \np@large.
169 \def\np@largeord#1#2#3{%
170 \ifcase#1 \or\or\or #3trent\or #3quarant\or #3cinqtant\or
171 #3sessant\or #3settant\or ottant\or #3novant\fi
172 \ifcase#2 \or un\or adu\or atre\or aquattr\or
173 acinqu\or asej\or asett\or ott\or anov\fi esim}

\np@hundredord \np@hundredord is similar to \np@hundred. It calls recursively \np@ordinall
with ‘o’ as second argument if # is not a multiple of 100.
174 \def\np@hundredord#1#2\np@del{%
175 \ifcase#1 \or\or due\or tre\or quattro\or cinque\or
176 sei\or sette\or otto\or nove\fi cent%
177 \ifnum#2=0 esim\else\np@ordinall{#2}{o}\fi}

\np@smallord The last three commands are similar to the similar uncapitalized ones; the last
\np@largeOrd calls \np@ordinall for the second step.
\np@hundredOrd 178 \def\np@smallOrd#1{%
179 \ifcase#1 \or \or \or \or \or \or \or \or \or \or \or \or \or
180 Undicesim\or Dodicesim\or Tredicesim\or Quattordicesim\or
181 Quindicesim\or Sedicesim\or Diciassettesim\or Diciottesim\or
182 Diciannovesim\or Ventesim\or Ventunesim\or Ventiduesim\or
183 Ventitreesim\or Ventiquattresim\or Venticinquesim\or
184 Ventiseiesim\or Ventisettesim\or Ventottesim\or Ventinovesim%
185 \fi}
186 \def\np@largeOrd#1#2{%
187 \ifcase#1 \or\or\or Trent\or Quarant\or Cinquant\or
188 Sessant\or Settant\or Ottant\or Novant\fi
189 \ifcase#2 \or un\or adu\or atre\or aquattr\or
190 acinqu\or asej\or asett\or ott\or anov\fi esim}
191 \def\np@hundredOrd#1#2\np@del{%
192 \ifcase#1 \or C\or Duec\or Trec\or Quattroc\or Cinquec\or
193 Seic\or Settec\or Ottoc\or Novec\fi ent%
194 \ifnum#2=0 esim\else\np@ordinall{#2}{o}\fi}
195 </package>

```

## Change History

```

v1.0
  General: Initial version . . . . . 1

```

## Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

	<b>Symbols</b>		
<code>\@Numeroinparole</code>	. 2, <u>24</u>	<code>\np@hundredord</code>	138, <u>181</u>
<code>\@Ordinalef</code>	..... 6, <u>119</u>	<code>\np@Large</code>	..... 68, <u>85</u>
<code>\@Ordinalem</code>	..... 5, <u>119</u>	<code>\np@large</code>	..... 52, <u>85</u>
<code>\@numeroinparole</code>	. 1, <u>24</u>	<code>\np@largeOrd</code>	.. 165, <u>185</u>
<code>\@ordialef</code>	..... 4, <u>119</u>	<code>\np@largeord</code>	.. 143, <u>176</u>
<code>\@ordinalem</code>	..... 3, <u>119</u>	<code>\np@Numeroinparole</code>	.
<code>\@printnumeroinparole</code>	..... 19, 21	.....	37, <u>39</u>
<code>\@printnumeroinparolef</code>	..... 21, 23	<code>\np@numeroinparole</code>	.
		.....	29,
		<code>\np@oa</code>	... 18, 23, 71, 88
	<b>N</b>	<code>\np@Ordinal</code>	121, 122, <u>123</u>
<code>\np@del</code>	..... 44,	<code>\np@ordinal</code>	119, 120, <u>123</u>
	47, 60, 63, 96,	<code>\np@Ordinall</code>	..... <u>123</u>
	101, 105, 111,	<code>\np@ordinall</code>	.....
	138, 160, 181, 200	.....	<u>123</u> , 184, 203
<code>\np@doprintdef</code>	8, 11–16	<code>\np@Small</code>	..... 66, <u>70</u>
<code>\np@Hundred</code>	..... 63, <u>96</u>	<code>\np@small</code>	..... 50, <u>70</u>
<code>\np@hundred</code>	..... 47, <u>96</u>	<code>\np@smallOrd</code>	.. 163, <u>185</u>
<code>\np@hundredOrd</code>	160, <u>185</u>	<code>\np@smallord</code>	.. 141, <u>167</u>
		<code>\np@Thousand</code>	... 60, <u>105</u>
		<code>\np@thousand</code>	... 44, <u>105</u>
		<code>\Numeroinparole</code>	.. <u>1</u> , <u>2</u>
		<code>\numeroinparole</code>	.. <u>1</u> , <u>2</u>
			<b>O</b>
		<code>\Ordinalef</code>	..... <u>1</u> , <u>2</u>
		<code>\ordialef</code>	..... <u>1</u> , <u>2</u>
		<code>\Ordinalem</code>	..... <u>1</u> , <u>2</u>
		<code>\ordinalem</code>	..... <u>1</u> , <u>2</u>
			<b>P</b>
		<code>\printNumeroinparole</code>	.....
		.....	3, <u>7</u>
		<code>\printnumeroinparole</code>	.....
		.....	3, <u>7</u>
		<code>\printOrdinalef</code>	.. 3, <u>7</u>
		<code>\printordialef</code>	.. 3, <u>7</u>
		<code>\printOrdinalem</code>	.. 3, <u>7</u>
		<code>\printordinalem</code>	.. 3, <u>7</u>