

# Manuale del pacchetto `liLaTeX`

Federico Miceli

estate 2024

## Indice

1	Introduzione	1
2	Come scaricare e inizializzare <code>liLaTeX</code>	2
3	Controllo correttezza inizializzazione	3
4	Com'è strutturato il file <code>liLaTeX</code>	3
5	FLAG	3
5.1	I valori possibili per la flag <code>\TipoDoc</code>	4
6	PACCHETTI	4
7	SOVRASCRITTURA DELLE FLAG	4
8	lett	5
9	AMBIENTI	5
9.1	Correzione della numerazione negli ambienti	5
9.2	L'ambiente <code>ArrowsEq</code>	6
10	MATHCAL E SIMILI	6
11	SI	6
12	ALTRE DEFINIZIONI UTILI	6
13	STRUMENTI PER GEOMETRIA	7
14	IMPAGINAZIONE	8
15	TABELLA VOTI	8

## 1 Introduzione

Questo manuale descrive il pacchetto `liLaTeX`. Si tratta di un pacchetto per  $\text{\LaTeX}$  pensato per i docenti di scuola superiore in Italia. Il prefisso `li` nel nome `liLaTeX` infatti rappresenta infatti la parola *liceo*. In questo senso `liLaTeX` è il pacchetto `LaTeX` per i docenti di liceo (o più in generale di scuole superiori italiane).

Il pacchetto è stato sviluppato da Federico Miceli, professore di matematica e fisica del Liceo Scientifico F.Vercelli di Asti, nel corso dell'estate 2024.

**Può essere liberamente scaricato (gratuitamente), utilizzato e modificato secondo le proprie esigenze.**

**È severamente vietato qualunque utilizzo commerciale e/o a scopo di lucro del pacchetto.**

## 2 Come scaricare e inizializzare liLaTeX

Il pacchetto liLaTeX è liberamente scaricabile alla pagina *gitHub* di Federico Miceli.

La pagina in questione può essere trovata all'indirizzo

<https://github.com/micelifrc/liLaTeX>

È possibile scaricare tutti i file all'interno della Repository, salvando tutti i file in una cartella denominata liLaTeX all'interno del proprio computer. In alternativa si può decidere di scaricare solo alcuni file fondamentali. All'interno della repository ci sono questi file:

- Il file `liLaTeX.tex` è il pacchetto vero e proprio. Devi scaricarlo all'interno della tua cartella liLaTeX.
- Il file `liLaTeXtikz.tex` è il pacchetto per le immagini. Devi scaricarlo e salvarlo all'interno della cartella liLaTeX. Se non ti interessa la parte grafica del pacchetto puoi evitare di scaricare questo file.
- I file `Manuale_liLaTeX.pdf` e `Manuale_liLaTeXtikz.pdf` contengono le istruzioni per i due diversi pacchetti (rispettivamente quello base e quello grafico).
- La cartella `Esempi` contiene alcuni semplici file creati con `liLaTeX.tex`. Non sono file fondamentali, ma può essere utili averli sott'occhio mentre impari a usare il pacchetto liLaTeX.tex
- La cartella `Immagini` contiene svariate immagini d'esempio create con `liLaTeXtikz.tex`. Anche questi file non sono essenziali, ma possono essere utili esempi di utilizzo del pacchetto grafico.
- Il file `README` contiene alcune informazioni di base sul pacchetto, ma se stai leggendo questo manuale l'avrai sicuramente già letto.

Una volta scaricati i file, è necessario scaricare anche il logo della propria scuola di riferimento, e salvarlo sul computer (generalmente all'interno della cartella liLaTeX).

Prima di poter utilizzare il pacchetto devi modificare le prime righe di codice del file `liLaTeX.tex`. Nelle prime righe del file sono infatti definiti alcuni parametri specifici per il modo in cui il file `liLaTeX.tex` è stato installato sul mio computer. In particolare devi modificare le seguenti righe:

1. `\providecommand{\liLaTeXPath}{C:/Users/themi/Desktop/LaTeX/liLaTeX}`  
Sostituire il percorso `C:/Users/themi/Desktop/LaTeX/liLaTeX` col percorso della propria cartella liLaTeX.
2. `\providecommand{\LogoScuolaPath}{C:/Users/themi/Desktop/LaTeX/liLaTeX/Logo_ls_Vercelli.jpeg}`  
Sostituire il percorso `C:/Users/themi/Desktop/LaTeX/liLaTeX/Logo_ls_Vercelli.jpeg` con quello del logo della propria scuola.
3. `\providecommand{\NomeScuola}{Liceo Scientifico \textit{F.Vercelli}}`  
Sostituire col nome della propria scuola.
4. `\providecommand{\CittaScuola}{Asti}`  
Sostituire `Asti` con la città della propria scuola.
5. `\providecommand{\AnnoScolastico}{2024-2025}`  
Questo idealmente va cambiato all'inizio di ogni anno scolastico.
6. `\providecommand{\PuntiBaseVerifica}{10}`  
Sostituisci il 10 col numero di punti di partenza generalmente dati nelle tue verifiche (nel mio caso sono 10/100). Se non dai punti di partenza cancella il 10.

Infine, se hai deciso di non scaricare il file `liLaTeXtikz.tex`, cerca la riga

`\providecommand{\inputTikz}{1}`

all'inizio del file `liLaTeX` e sostituisci l'1 con uno 0.

### 3 Controllo correttezza inizializzazione

Nella cartella **Esempi** della repository di gitHub è presente un file chiamato

`verifica_controllo_installazione.tex`

Prova a scaricarlo e a compilarlo.

Dovresti ottenere un file analogo a quello chiamato `verifica_controllo_installazione.pdf` (sempre nella cartella **Esempi**).

Se la tua scuola ha un nome molto lungo potresti avere alcuni problema con la tabella a inizio file. Nel caso, cerca la riga

`\pgfmathsetmacro\NomeCognomeWidthDiffInTab{5}`

all'interno del file `liLaTeX` e prova a sostituire il numero 5 con altri valori numerici. Quello è il numero che controlla le proporzioni fra le due colonne nella tabella a inizio file nelle verifiche.

### 4 Com'è strutturato il file `liLaTeX`

Il file `liLaTeX` è diviso in diverse sezioni, ciascuna delle quali è più o meno monotematica. Le sezioni sono:

1. FLAG
2. PACCHETTI
3. SOVRASCRITTURA DELLE FLAG
4. lett
5. AMBIENTI
6. MATHCAL E SIMILI
7. SI
8. ALTRE DEFINIZIONI UTILI
9. STRUMENTI PER GEOMETRIA
10. IMPAGINAZIONE
11. TABELLA VOTI

Infine, nell'ultima riga del file, viene importato il file `liLaTeXtikz.tex`.

In questo manuale analizziamo una sezione per volta, descrivendo le funzioni principali di ciascuna di queste sezioni.

### 5 FLAG

La prima sezione del file `liLaTeX` definisce una serie di FLAG. Oltre a quelle corrette in fase di inizializzazione (già descritte nella sezione 2), sono presenti le seguenti flag, che verranno specificate da un file all'altro:

- `\Data`, la data in cui verrà utilizzato il file;
- `\NomeStudente` o `\CognomeStudente`, specifica il nome e il cognome dello studente a cui è destinato il file (se è destinato a uno studente nello specifico). Se il file è destinato all'intera classe conviene lasciarlo vuoto;
- i campi `\Materia` e `\Minuti` sono utili per le verifiche. Specificano la materia della verifica e il numero di minuti di durata della stessa;
- `\Titolo` il titolo del file. Se non specificato, viene inizializzato automaticamente per le verifiche;
- `\MarginSize` e `\FontSize` sono le dimensioni dei margini (in **cm**) e del testo (in **pt**, che è una misura interna di  $\text{\LaTeX}$ ). `\MarginSize` è normalmente uguale a 1.5, mentre `FontSize` è normalmente 11. Considera comunque che `\MarginSize` viene ignorato (e azzerato) se `\TipoDoc` è *immagine*;
- `\colText`, `\colSol` e `\colSI` sono rispettivamente il colore con cui scriviamo i testi degli esercizi (o gli enunciati dei problemi), il colore con cui scriviamo le soluzioni dei problemi (o le dimostrazioni dei problemi) e il colore che usiamo per le unità di misura. Normalmente `\colText` è **black**, `\colSol` è **blue**, e `\colSI` è un arancione-rame (viene però cambiato in **black** per i `\TipoDoc` che in genere vengono stampati);
- `\inputTikz` può avere valore 0 (falso) oppure 1 (vero), e dice se vogliamo importare anche il file `liLaTeXtikz.tex`;

- `\EnumPages` può avere valore 0 (falso) oppure 1 (vero). Dice se vogliamo mettere i numeri di pagina nell'impaginazione. Viene resettata come immagine se `\TipoDoc` è uguale a `immagine` (è l'unico caso in cui una flag viene sovrascritta anche se specificata dall'utente);
- `\TipoDoc` è il tipo di documento che vogliamo produrre. Sono supportati i tipi di documenti specificati in seguito in questa sezione.

Per modificare una flag, è necessario ridefinirla nel file prima in scrivere l'`\input` di `liLaTeX.tex`.

Ad esempio se vogliamo scrivere che stiamo scrivendo una verifica di fisica, nel preambolo del file (dopo la `\documentclass` ma prima dell'`\input` di `liLaTeX.tex`) scriveremo le due righe:

```
\newcommand{\TipoDoc}{verifica}
\newcommand{\Materia}{Fisica}
```

## 5.1 I valori possibili per la flag `\TipoDoc`

La flag `\TipoDoc` indica il tipo di documento che stiamo scrivendo. Al momento sono implementati i seguenti tipi di file (anche se non sono del tutto diversi fra loro, al momento):

- `plain` se è un documento senza nulla di particolare (è il valore di default);
- `libro` per testi molto lunghi (come questo manuale);
- `immagine` se vogliamo semplicemente produrre un'immagine (utile per il pacchetto `liLaTeXtikz.tex`);
- `verifica` per le verifiche;
- `verifica recupero` per le verifiche di recupero del debito (del primo o del secondo periodo);
- `esercizio svolto` per esercizi svolti che vogliamo condividere con gli studenti;
- `lista esercizi` per scrivere una lista di esercizi da assegnare (ad esempio per le vacanze estive);
- `scheda` per una scheda di lavoro;
- `scheda SSPM` per le schede di SSPM (*Scuole Secondarie Potenziate in Matematica*, comunemente chiamato *Liceo Matematico*);
- `scheda laboratorio` per le schede di laboratorio di fisica;
- `prova laboratorio` per le prove (valutate) in laboratorio di fisica;
- `gara olimpiadi` per scrivere il testo di una gara (o una simulazione di gara) per le olimpiadi di matematica (me ne occupo personalmente al Vercelli di Asti);
- `lezione` per scrivere una lezione o un'UDA (più o meno approfondita). Può poi essere condivisa con gli studenti, o può essere utile a noi come appunti se facciamo una lezione un po' particolare che vogliamo ricordare negli anni successivi o che vogliamo condividere coi colleghi;
- `dimostrazione` la dimostrazione di un teorema (o di una lista di teoremi);
- `formulario` può essere utilizzata per scrivere piccoli formulari, eventualmente da condividere con gli studenti.

La lista potrà ovviamente essere estesa in futuro, man mano che il progetto `liLaTeX` viene ampliato.

## 6 PACCHETTI

In questa sezione vengono importati tutti i pacchetti esterni. Si tratta di una lista relativamente lunga, ma senza nulla di particolare da dire.

Se l'utente deve aggiungere un pacchetto dovrà di volta in volta valutare se aggiungerlo direttamente al file `liLaTeX.tex` (se si tratta di un pacchetto che intende utilizzare spesso) oppure se scriverlo solo nel file in cui è utile. Nel secondo caso si suggerisce di includere il pacchetto *dopo* aver importato `liLaTeX.tex` per ridurre il rischio di conflitti fra pacchetti.

**Attenzione!** Evita di spostare i pacchetti all'interno del file `liLaTeX.tex`. Infatti alcuni pacchetti possono entrare fra loro in conflitto se vengono scambiati.

## 7 SOVRASCRITTURA DELLE FLAG

Alcune flag vengono inizializzate in un secondo momento (se non sono state inizializzate dall'utente all'interno del file).

In particolare la flag `\Titolo` viene inizializzata automaticamente nel caso di verifiche (a meno che sia stata fornita dall'utente).

## 8 lett

L'ambiente `lett` viene usato in modo analogo ad `enumerate` per le liste puntate. La differenza è che indicizza con le lettere anziché coi numeri.

È possibile specificare le opzioni `[noitemsep]` e `[nolistsep]` (come in `enumerate`).

Esiste anche l'opzione `lettfrom` che ci permette di ricominciare l'indicizzazione da un punto successivo.

Per esempio, consideriamo questo codice, che produce questo output (l'output è colorato in blue):

```
\begin{lett}
\item Prima riga
\item Seconda riga
\end{lett}
Testo intermedio
\begin{lettfrom}{3}
\item Terzo punto
\end{lettfrom}
```

- a. Prima riga
- b. Seconda riga

Testo intermedio

- c. Terzo punto

## 9 AMBIENTI

Gli ambienti possono essere utilizzati per scrivere esercizi, teoremi, dimostrazioni, soluzioni e così via.

Per scrivere un esercizio usiamo la sintassi (il titolo è facoltativo)

```
\begin{Ex}[titolo]...\end{Ex}
```

Esistono i seguenti ambienti definiti nel pacchetto `liLaTeX.tex`:

- `Ex` per creare un esercizio;
- `Exsec` analogo ad `Ex`, ma indicizzato sulle sezioni;
- `Probl` per creare un problema (simile ad `Ex`);
- `Def` per dare una definizione;
- `teo` per creare un teorema;
- `lemma` per creare un lemma;
- `cor` per scrivere un corollario;
- `Sol` per scrivere la soluzione di un esercizio;
- `Dim` per dare la dimostrazione di un teorema (o risolvere un esercizio dimostrativo).

Per la maggior parte di questi ambienti esiste anche una versione equivalente con l'asterisco se non vogliamo numerarli. Ad esempio possiamo usare `Def*` se non vogliamo dare una numerazione alla definizione.

### 9.1 Correzione della numerazione negli ambienti

A volte potremmo trovare un ambiente numerato in modo *sbagliato* (nel senso che non è quello che vorremmo). In questi casi è possibile cambiare la numerazione dell'ambiente utilizzando il comando `\setCounter` (da usare subito prima di usare l'ambiente).

Ad esempio, se vogliamo resettare la numerazione di `\Sol` in modo che il prossimo ambiente `\Sol` abbia numero 5 scriveremo il comando `\setCounter{Sol}{5}`, come qui illustrato (il testo grigio produce l'output blu):

```
\setCounter{Sol}{5}
\begin{Sol}
\Questa è la soluzione del problema.
\end{Sol}
```

**Soluzione 5.** Questa è la soluzione del problema.

## 9.2 L'ambiente ArrowsEq

L'ambiente `ArrowsEq` può essere utilizzato per scrivere un'equazione con delle frecce fra una riga e la successiva, per indicare cosa abbiamo fatto in ciascun passaggio.

La sua sintassi è analoga a quella di `align*`. Quando andiamo a capo però, anziché usare il comando `\\`, possiamo usare il comando `\Arrowline[col]{...}` in cui spieghiamo che cosa abbiamo fatto. Considera il seguente esempio (codice in grigio, output in blu):

```
\begin{ArrowsEq}
  &x=x+2\Arrowline{-x}
  &0=2
\end{ArrowsEq}
```

$$\begin{array}{l} x = x + 2 \\ 0 = 2 \end{array} \quad \begin{array}{l} \curvearrowright \\ \curvearrowright \end{array} -x$$

## 10 MATHCAL E SIMILI

Per scrivere le lettere in ambienti `\mathcal`, `\mathrm` e `\mathbb` utilizziamo dei comandi abbreviati.

Così un comando come `\rs` scrive la lettera **s** in `\mathrm`, mentre il comando `\rA` scrive la lettera **A** in `\mathrm`. In generale, quindi, per scrivere una lettera in `\mathrm` ci basta scrivere la lettera preceduta da `\r`.

In modo analogo, per scrivere una lettera (maiuscola) in `\mathcal` o in `\mathbb` basta scrivere, rispettivamente, `\c` o `\b` prima della lettera. Per il `\mathbb` c'è anche la possibilità di scrivere un 1, usando il comando `\bUNO`.

Ci sono poi i comandi `\i` e `\j` che producono le lettere *i* e *j* senza il puntino.

Infine, possiamo scrivere le lettere in rosso (o nel colore specificato con la flag `\colInc`) per indicare che le lettere così rappresentate sono incognite (unknown). Per farlo, prima della lettera scriviamo `\u`.

Questo è l'elenco dei caratteri che si ottiene in questi modi:

<b>ABCDEFGHIJKLMN</b>	<b>OPQRSTUVWXYZ</b>	<b>abcdefghijklmnopqrstuvwxyz</b>
ABCDEFGHIJKLMN	OPQRSTUVWXYZ	abcdefghijklmnopqrstuvwxyz
<i>ABCDEFGHIJKLMN</i>	<i>OPQRSTUVWXYZ</i>	ABCDEFGHIJKLMN <b>OPQRSTUVWXYZ1</b>

## 11 SI

Le unità di misura vengono usate particolarmente spesso in questi file. Per scriverle possiamo usare i comandi `\SI{...}` che scriverà le unità di misura nel colore corretto. Se l'unità di misura si presenta come una frazione possiamo invece usare il comando `\SIf{numeratore}{denominatore}`.

C'è infine il caso dei gradi Celsius, che vengono scritti col comando `\oC`.

Un comando come `\SI{J/\oC kg}` produce quindi il risultato **J/°Ckg**, mentre il comando `\SIf{m}{\s^2}` produce il risultato  $\frac{\text{m}}{\text{s}^2}$ .

Il colore utilizzato sarà quello specificato tramite la flag `\colSI` (nero per file da stampare, ramato per quelli da lasciare solo come pdf).

## 12 ALTRE DEFINIZIONI UTILI

In questa sezione sono presenti comandi vari che possono risultare utili per varie ragioni.

Il primo comando interessante è `\link[colore]{link}` che genera un link del colore specificato (blu di default) che porta all'indirizzo specificato con l'input `link`.

Il comando `\riga` tira una riga orizzontale nel foglio.

C'è poi il comando `\pare[L]{INT}` che crea delle parentesi intorno ad `INT` del tipo precisato con l'input

L. Le parentesi supportate sono:

- `L=t` (default) parentesi tonde;
- `L=q` parentesi quadre;
- `L=g` parentesi graffe;

- L=v parentesi verticali (ad esempio per il valore assoluto);
- L=f floor (arrotondamento per difetto);
- L=c ceil (arrotondamento per eccesso).

Gli output che si ottengono sono quindi

$$\left(\int e^{-x^2} dx\right) \quad \left[\int e^{-x^2} dx\right] \quad \left\{\int e^{-x^2} dx\right\} \quad \left|\int e^{-x^2} dx\right| \quad \left\lfloor\int e^{-x^2} dx\right\rfloor \quad \left\lceil\int e^{-x^2} dx\right\rceil$$

Il comando `\ang{A}{B}{C}` crea un angolo  $\widehat{ABC}$ .

Il comando `\bin{n}{k}` crea un binomiale  $\binom{n}{k}$ .

I comandi `\vc{v}` e `\vcp{w}{1}` creano i due vettori  $\vec{v}$  e  $\vec{w}$ .

Il comando `\vett{a}{b}{c}` crea il vettore  $\begin{pmatrix} a \\ b \\ c \end{pmatrix}$ . Questo comando può prendere un numero arbitrario di

input (a discrezione dell'utente).

I comandi `\seq`, `\scong` e `\lr` creano dei simboli ( $=$ ,  $\cong$  e  $\implies$ ) sovrastati dall'argomento opzionale. C'è anche la versione `\lrt` analogo di `\lr`, in cui il commento è scritto fra `\text` (gli altri tre hanno commenti in ambiente matematico).

Quindi i comandi `\lrt[a]`, `\lr[a]`, `\seq[x]` e `\scong[y]` producono gli output  $\overset{a}{\implies}$ ,  $\overset{a}{\implies}$ ,  $\overset{x}{=}$  e  $\overset{y}{\cong}$ .

Spesso dei comandi analoghi vengono usati come separatori. I più comuni sono `\qe` e `\qlr` (o i loro analoghi `\qqe` e `\qqlr`, in cui viene lasciato maggior spazio). Ad esempio il comando in grigio genera l'output blu:

`x \qe y \qqlr[a] z \qlr t`

$$x \text{ e } y \overset{a}{\implies} z \implies t$$

## 13 STRUMENTI PER GEOMETRIA

È comune in problemi di geometri (ma non solo) chiedere esplicitamente (nel testo) di impostare un sistema per risolvere il problema (altrimenti gli studenti potrebbero trovare la soluzione a occhio, e poi lamentare il fatto che non hanno ottenuto i punti del problema, quando tecnicamente hanno trovato la soluzione corretta, e magari l'hanno pure verificata). Per scongiurare questo rischio, in questi problemi conviene usare il comando `\eqreq` che produce il seguente output (in rosso):

**Attenzione!** Per risolvere il problema devi impostare e risolvere un'opportuna equazione o un sistema di equazioni. Soluzioni trovate "a tentativi" e solo verificate non sono considerate accettabili.

Spesso nelle dimostrazioni di geometria dobbiamo fare delle implicazioni. Per farlo possiamo usare il comando `\Impl[nome]{Hp1}{Hp2}...\{Th}` (possiamo usare quante ipotesi vogliamo).

Ad esempio il seguente comando (grigio), fuori dall'ambiente matematica, crea il seguente output (blu):

```
\Impl[crit.parall.]{
  {\ang{A}{B}{C}\cong\ang{D}{C}{B}}
  {\$BC\$ trasversale comune}
}{\$AB \$parallel CD\$}
```

$$\left. \begin{array}{l} \widehat{ABC} \cong \widehat{DCB} \\ BC \text{ trasversale comune} \end{array} \right\} \overset{\text{crit.parall.}}{\implies} AB \parallel CD$$

Dal momento che i criteri di congruenza e di similitudine sono particolarmente comuni, ho scritto anche dei comandi specifici per quelle due famiglie di criteri.

Per i criteri di congruenza la sintassi è `\cc[nome]{tr1}{tr2}{Hp1}{Hp2}{Hp3}` dove `nome` è il nome del criterio utilizzato, `tr1` e `tr2` sono i due triangoli fra loro congruenti e `HP1`, `HP2`, `HP3` sono le ipotesi confrontate. Ad esempio, il testo in blu viene generato col comando grigio:

```
\cc[3°o.c.]{ABC}{A'B'C'}{\$AB \$cong A'B'$}{\$BC \$cong B'C'$}{\$CA \$cong C'A'$}
```

Consideriamo i triangoli  $ABC$  e  $A'B'C'$ :

$$\left. \begin{array}{l} AB \cong A'B' \\ BC \cong B'C' \\ CA \cong C'A' \end{array} \right\} \overset{3^\circ \text{o.c.}}{\implies} ABC \cong A'B'C'$$

**Attenzione!** Osserva che le ipotesi vengono scritte in ambiente matematico, ma che i triangoli vengono scritti in testo semplice.

Per i criteri di similitudine la sintassi è identica (si chiama `\cs` anziché `\cc` e usa 2 ipotesi anziché 3).

## 14 IMPAGINAZIONE

L'impaginazione cambia a seconda del tipo di documento (specificato con la flag `\TipoDoc`) e dal valore 0 – 1 dato alla flag `\EnumPages` (vero di default, a meno che `\TipoDoc=immagine`), oltre ovviamente alla misura dei margini, specificata con la flag `\MarginSize`.

L'impaginazione utilizzata sarà `empty` oppure `plain` (a seconda del valore di `\EnumPages`).

Se però il `TipoDoc` è specifico di una verifica, l'impaginazione è molto più elaborata.

Nel caso, l'impaginazione aggiungerà (in cima alle pagine dispari) un titolo (specificato nella flag `\Titolo`, ma eventualmente arricchito con `\Minuti`).

Poi verrà creata una tabella in cui gli studenti inseriscono i propri dati. Qui compare anche il logo della scuola. Si suggerisce di consultare il file `\verifica_controllo_installazione.pdf` (a cui abbiamo già fatto riferimento nella sezione 3), per un esempio sul suo funzionamento.

## 15 TABELLA VOTI

La tabella dei voti deve essere scritta in ogni verifica (generalmente alla fine della verifica).

La sua sintassi è del tipo `\TabellaVoti{{Es1}{Es2}...}` dove ogni esercizio può a sua volta essere separato in vari punti.

Ad esempio, se abbiamo una verifica in 5 esercizi, ognuno dei quali vale 18 punti (e 10 punti base, specificati con la flag `\PuntiBaseVerifica`) possiamo generare la tabella dei voti col comando

`\TabellaVoti{{18}{18}{18}{18}{18}}`, che genera questa tabella:

Pt.base	Es.1	Es.2	Es.3	Es.4	Es.5	Tot.
10	18	18	18	18	18	100
10						

**Voto:**

Se invece il terzo esercizio è diviso in due parti (ciascuna delle quali vale 9 punti) e il quinto esercizio è diviso in tre parti (ciascuna delle quali vale 6 punti) la tabella dei voti viene generata col comando `\TabellaVoti{{18}{18}{9,9}{18}{6,6,6}}`, che genera questa tabella (automaticamente messa a fine pagina):

Pt.base	Es.1	Es.2	Es.3		Es.4	Es.5			Tot.
			a.	b.		a.	b.	c.	
10	18	18	9	9	18	6	6	6	100
10									

**Voto:**