

AcroT<sub>E</sub>X.Net

## PDF Flash Card: Arithmetic

D. P. Story

## Table of Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Requirements and Options</b>	<b>4</b>
2.1	Requirements . . . . .	4
2.2	Options . . . . .	4
<b>3</b>	<b>The Components of the Flash Card</b>	<b>5</b>
3.1	Settings . . . . .	5
•	Options . . . . .	5
•	Toggle Keypad . . . . .	7
•	About PDF Flash Card . . . . .	7
3.2	The <code>\arithProb</code> Command . . . . .	7
3.3	The <code>\inputRegion</code> Command . . . . .	9
3.4	The <code>\startAgain</code> and <code>\newCard</code> Command . . . . .	10
3.5	The <code>\alertbox</code> Command . . . . .	10
3.6	The <code>\Keypad</code> Command . . . . .	11
3.7	The <code>\cbTiming</code> Command . . . . .	11
3.8	The <code>\ansField</code> Command . . . . .	12
3.9	The <code>\cbOperation</code> Command . . . . .	13
3.10	The <code>\statsFields</code> Command . . . . .	13
<b>4</b>	<b>Other Customizations</b>	<b>13</b>
4.1	Setting the Range and Decimal . . . . .	14
4.2	Customization Strings . . . . .	15
<b>5</b>	<b>Suggested Layout</b>	<b>16</b>

## 1. Introduction

The `fc_arith` package is used to create an electronic “flash card” used to drill a student on elementary arithmetic, addition, subtraction, multiplication, and division. There are options for setting the range of the numbers to be used to randomly select numbers that appear in the arithmetic problem. Numbers can be set to 0, 1, or 2 decimal places. There is an optional timing mechanism that can be used to test a student’s quickness in solving problems. There is a collection of fields used to tally the student’s work on the arithmetic problems.

**History.** This is a re-work of an earlier PDF on drilling arithmetic problems. At that time, I took a blank PDF page and used the user interface of Acrobat to build the flash card.

In this version, I’ve taken the original flash card and developed a  $\LaTeX$  package for generating the flash card, added a number of customization options, and many other enhancements.

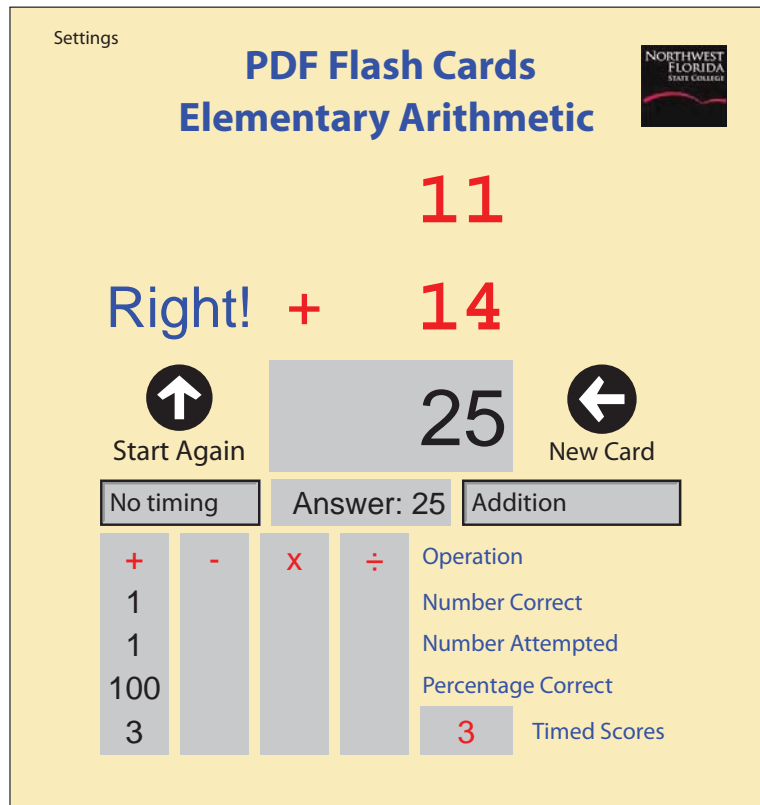


Figure 1: PDF Flash Card: Arithmetic

## 2. Requirements and Options

In this section, we review the requirements to build your own custom flash card, and describe the package options available.

### 2.1. Requirements

The package is designed to be used with either a Acrobat Distiller or a pdf<sub>l</sub>atex workflow. The only true requirements are

- The eforms package (version 2.5c or later, 2010/03/21 or later), a package distributed with AeB (the Acro<sub>T</sub>E<sub>X</sub> eDucation Bundle<sup>1</sup>)
  - The popupmenu package<sup>2</sup> which creates a menu system.
- ▶ Sample files: fc04.tex and fc04\_pdf<sub>l</sub>atex.tex.

I recommend the use of the web package to design your page, just as the sample files do. Enhancements include the use of the aeb<sub>pro</sub> package; see the sample file fc04.tex for an illustration of the use of aeb<sub>pro</sub>.

### 2.2. Options

1. allownegsub: The original arithmetic flash card did not allow a negative difference, if this option is used, the differences are allowed to be negative. The default is to not to allow negative differences.
2. nomenu: There is a menu, labeled **Settings**, that allows the user change the range of the numbers generated by the flash card, set the number of decimal places, allows the user to toggle the **Keypad** on and off, and so on. For further details see 'Settings' on page 5.  
If this option is used, the menu does not appear. The default is for it to appear.
3. showkeypadlink: The showkeypadlink generates a link with a label of **Toggle Keypad** just above the keypad. Clicking the link toggles the visibility of the keypad.  
This option is useful if the nomenu option is taken, but you want the keypad available to users.
4. operations: Gives the ability to declare what operations flash card should show. Supported operations are add, sub, mul, div. For example,

```
operations={add,sub}
```

creates a flash card in which only addition and subtraction problems are available. The default is to make available all operations.

<sup>1</sup><http://www.math.uakron.edu/~dpstory/webeq.html>

<sup>2</sup><http://mirror.ctan.org/macros/latex/contrib/popupmenu/>

5. `notimescores`: As can be seen in [Figure 1](#), page 3, the flash card has many fields that key of the user's performance. One of these is the **Timed Scores** row; this row gives a score based on time required to answer the problem correctly. Some teachers may not want such pressure placed on their students, so if this option is taken, the row is not generated.

### 3. The Components of the Flash Card

We describe the elements of the flash card as they appear, from top to bottom, in [Figure 1](#), page 3.

#### 3.1. Settings

The **Settings** menu, which can be removed with the `nomenu` option, consists of three submenu items: **Options**, **Toggle Keypad**, and **About PDF Flash Cards**, see the figure below.

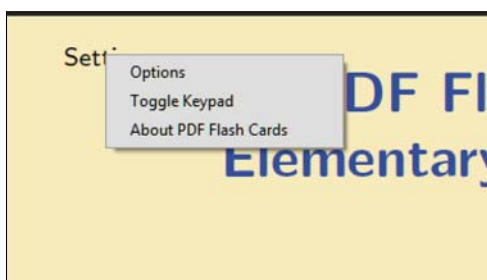


Figure 2: The **Settings** menu

The labeling for the **Settings** menu is modified using the command `\fcSettings`, the default definition is `\newcommand{\fcSettings}{Settings}`. The color of the link is set by the command `\fcSettingsColor`, the default definition of which is `\newcommand{\fcSettingsColor}{black}`.

#### • Options

The **Options** submenu item calls forth the **Options** dialog box, [Figure 3](#), page 6. Through this dialog box, you can set the intervals from which random numbers are drawn, and their precision.

**Addition, Subtraction, Multiplication.** We handle these three operations in the same way, so we explain them together. For these three we have a top number and a bottom number.

$$\begin{array}{r} \text{top} \quad 14 \quad 17 \quad 12 \\ \text{bottom} \quad + \quad 2 \quad - \quad 4 \quad \times \quad 7 \end{array}$$

In the dialog box, you can set the range (the interval) of numbers from which randomly generated numbers are taken. For example, the [Figure 3](#) on page 6 specifies that for Addition problems, the top value should come from a range of 2 to 20, the bottom range is 5 to 3, and there should be no decimal places (hence, for this example, we deal with integer arithmetic).

The PDF Flash Card supports at most two decimal places.

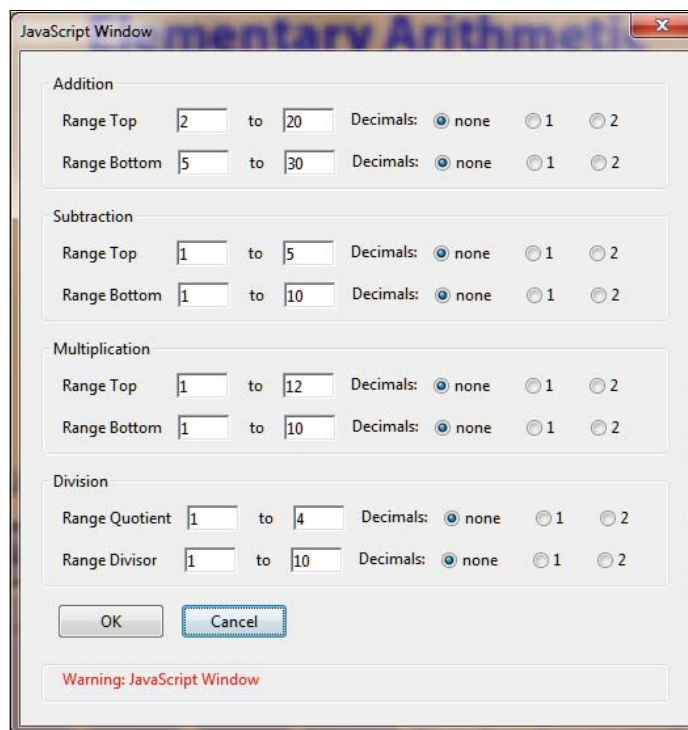


Figure 3: The **Options** dialog box

**Division.** Division is handled separately because of the way the problem is generated. Rather than randomly generating the top and bottom values as is done for addition, subtraction, and multiplication, the `fc_arith` package randomly generates the *quotient* and *divisor*, then calculates the dividend

$$\text{dividend} = \text{quotient} \times \text{divisor}$$

This leads to the assurance of a clean division problem with no nasty rounding, or infinite decimal expansions ( $1/30.33333\dots$ ). It is a design decision I made for the original PDF Flash Card; the targeted audience was student in elementary school.

The top is now the dividend and the bottom is the divisor, but in the **Options** dialog box, **Range Top** and **Range Bottom** is replaced by **Range Quotient** and **Range Divisor** to reflect the different way of generating the division problem.

► The dialog box retains the settings only during the current session of Adobe Reader. The Reader does not have the ability to save data so once the Flash Card is closed, the data is lost.

#### • Toggle Keypad

Selecting the **Toggle Keypad** submenu item toggles the keypad, which is initially hidden, see [Figure 4](#). The keyboard input region is changed to read only, the student can only enter through the keypad.

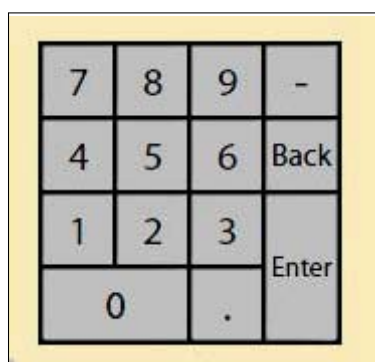


Figure 4: The Keypad

#### • About PDF Flash Card

Some information about the PDF Flash Card.

### 3.2. The `\arithProb` Command

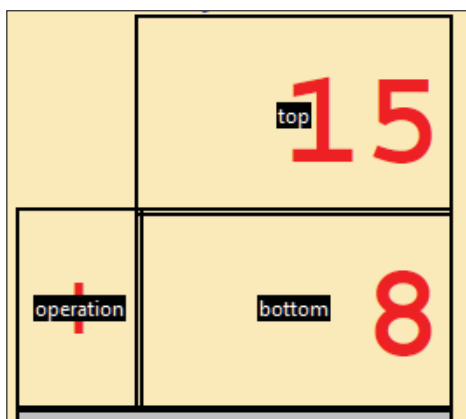
The `\arithProb` command is a trio of form fields laid out to display an arithmetic problem, see [Figure 5](#),<sup>3</sup> and [Figure 6](#), page 9.

The three fields and their descriptions follow:

- **top**: The name of this text field is `top`, and it holds the upper most number in the problem (this number is called by different names depending on the operation, we'll just call it the top number).
- **bottom**: The name of this text field is `bottom`, and it holds the lower number in the problem.
- **operation**: The name of this text field is `operation`.

The top and bottom fields use a monospace font (courier-bold) so that when there are decimal numbers involved, the numbers will align properly.

<sup>3</sup>Figure 5 shows an outline of the fields so you can see their relative positions.

Figure 5: The `\arithProb` Command

**Appearance Parameters for `\arithProb`.** The three fields are bundled together as a unit, which makes it difficult to set the appearance of each individual field, unless you want to redefine `\arithProb`. There are, however, several commands available to make some adjustments to meet your needs.

- `\tBGNoBorder`: This is a command that holds various eform key-value pairs that are passed to these three fields. The default definition of `\tBGNoBorder` is

```
\newcommand{\tBGNoBorder}{\BC{}\BG{}\autoCenter{n}
\textSize{0}\textColor{1 0 0}\Ff\FfReadOnly}
```

This set of parameters gives a field with transparent border and background, auto-adjusting font size, red text, and with a readonly attribute. This command can be redefined.

- `\monoSpaceFont`: These three fields use a monospace font, you can change this font using `\monoSpaceFont`. The default definition is

```
\newcommand{\monoSpaceFont}{CoBo}
```

which is courier-bold.

- `\setDimOf`: The `\setDimOf` can be used to set the dimensions of these fields. The command takes three parameters, field name, width and height. The default definitions for these three fields are

```
\setDimOf{top}{1in}{0.62in}
\setDimOf{operation}{.38in}{.62in}
\setDimOf{bottom}{1in}{0.62in}
```

Note that the dimensions of `top` and `bottom` are the same, as they are supposed to align vertically.



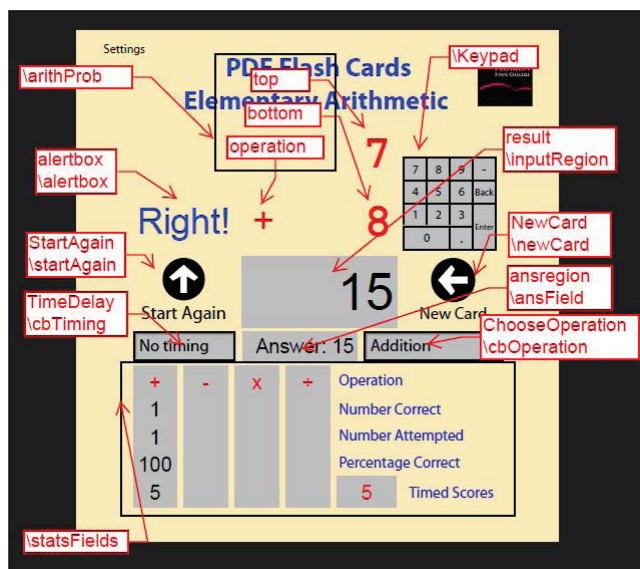


Figure 6: The PDF Flash Card, Annotated

### 3.3. The `\inputRegion` Command

When the keypad field is hidden (see the ‘[The `\Keypad` Command](#)’ on page 11), the user inputs the response to the arithmetic problem into the field titled `result`. This field is placed on the flash card using the `\inputRegion` command; this field is usually placed immediately beneath the `\arithProb` command so that the `result` field is aligned vertically with the `top` and `bottom` fields.

The syntax for `\inputRegion` is

```
\inputRegion[<eform_parameters>]
```

The default appearance properties are pre-defined as

```
\newcommand{\cBGNoBorder}{\Q{1}\BC{} \BG{} \autoCenter{n}
\textSize{0}\textColor{0 0 0}\BG{.75 .75 .75}
\Ff\FfReadOnly}
```

You can redefine `\cBGNoBorder`, or you can introduce a few changes through the optional parameter.

The dimensions of the `result` fields can be set using `\setDimOf`. The default definition is

```
\setDimOf{result}{1.38in}{0.62in}
```

The `0.62in` is the same width used for the `top` and `bottom` fields.

### 3.4. The `\startAgain` and `\newCard` Command

The `\newCard` command generates a push button named `NewProblem`, when pushed, a new arithmetic is randomly generated. The `\startAgain` command creates a push button named `StartAgain`, when pushed, all fields are cleared, and any JS variables are re-initialized. The syntax for these two fields is

```
\startAgain[<eform_parameters>]
\newCard[<eform_parameters>]
```

The default appearance properties are pre-defined as

```
\newcommand{\tBGNoBorderI}{\BC{}\BG{}\autoCenter{n}
\textSize{0}\textColor{1 0 0}}
```

You can redefine `\tBGNoBorderI`, or you can introduce a few changes through the optional parameter.

The dimensions of `NewProblem` and `StartAgain` can be set using `\setDimOf`. The default definition is

```
\setDimOf{StartAgain}{0.88in}{0.62in}
\setDimOf{NewProblem}{0.88in}{0.62in}
```

The text font used by these two fields is determined by the command `\fieldFont`. The default definition of `\fieldFont` is

```
\newcommand{\fieldFont}{Helv}
```

The name for the font should be one of the 13 basic fonts, or a PostScript font. In the latter case, most likely, the Acrobat Distiller is needed for PDF creation.

### 3.5. The `\alertbox` Command

The `\alertbox` command creates a transparent text field named `alertbox`, which displays **Right!** and **Wrong!** messages. The syntax for this field is

```
\alertbox[<eform_parameters>]
```

The default appearance properties are pre-defined as

```
\newcommand{\tBGNoBorder}{\BC{}\BG{}\autoCenter{n}
\textSize{0}\textColor{1 0 0}\Ff\FfReadOnly}
```

You can redefine `\tBGNoBorder`, or you can introduce a few changes through the optional parameter.

The dimensions of `alertbox` can be set using `\setDimOf`. The default definition is

```
\setDimOf{alertbox}{.88in}{.62in}
```

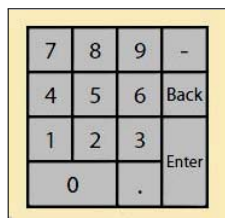


Figure 7: The Keypad

### 3.6. The `\Keypad` Command

The `\Keypad` creates a number of fields in the form of a keypad. See Figure 3.6. When the keypad is visible, the result field (created by `\inputRegion`) is read-only.

The default appearance properties are pre-defined as

```
\newcommand{\tBGNoBorder}{\BC{\BG}\autoCenter{n}
\textSize{0}\textColor{1 0 0}\Ff\FfReadOnly}
```

You can redefine `\tBGNoBorder`, or you can introduce a few changes through the optional parameter.

The dimension of each key is based on the value of `\szNum`, the default definition is

```
\newcommand{\szNum}{14bp}
```

The default appearance properties are pre-defined as

```
\newcommand{\myNumPadI}{\F\FHhidden\autoCenter{n}
\textSize{8}\textFont{\fieldFont}\S{S}}
```

You can redefine `\myNumPadI`.

The labeling for the **Enter** and **Back** keys can be redefined as well, their definitions are

```
\newcommand{\kpBack}{Back}
\newcommand{\kpEnter}{Enter}
```

### 3.7. The `\cbTiming` Command

The `\cbTiming` command creates a combo box, named `TimeDelay`, that lists a menu of time values for the user to practice against. The default is **No Timing**.

**Rules for Timed Responses.** When **No Timing** is in effect, if the student correctly answers the problem in the time allotted, credit is awarded; otherwise, no credit is given. When another time setting is in effect, the same rule applies as above, except no credit is awarded if the time is up, even if the response is correct. Click on the push button labeled **Timed Scores** to see the rules for awarding the points.

**Note:** The original idea behind the timed scores is to have students compete against each other. For example, for addition, let each student attempt 10 problems, the highest

score for wins! Try this with **No Timing** first, then start decreasing the time value progressively down to 5 sec. Who is the fastest with the mostest!

The syntax for `\cbTiming` is

```
\cbTiming[<eform_parameters>]
```

The default appearance properties are pre-defined as

```
\newcommand{\cBGBorder}{\BC{0 0 0}\BG{.75 .75 .75}
\autoCenter{n}\textSize{0}\textColor{0 0 0}}
```

You can redefine `\cBGBorder`, or you can introduce a few changes through the optional parameter.

The dimensions of `TimeDelay` field can be set using `\setDimOf`:

```
\setDimOf{TimeDelay}{0.9in}{0.24in}
```

The font used in the list box is determined by the value of `\fieldFont`, the default definition is `\newcommand{\fieldFont}{Helv}`.

The string **No Timing** can be modified using `\fcNoTiming`:

```
\newcommand{\fcNoTiming}{No Timing}
```

### 3.8. The `\ansField` Command

The `\ansField` creates a text field named `ansregion`; the field holds the correct answer to the problem.

The syntax for `\cbTiming` is

```
\ansField[<eform_parameters>]
```

The appearance of this field can be modified using `\cBGNoBorder`; the default definition is,

```
\newcommand{\cBGNoBorder}{\Q{1}\BC{} \BG{} \autoCenter{n}
\textSize{0}\textColor{0 0 0}\BG{.75 .75 .75}\Ff\FfReadOnly}
```

This can be redefined as desired.

The dimensions of `ansregion` can be set be `\setDimOf`:

```
\setDimOf{ansregion}{.87in+10bp}{.24in}
```

Notice that a little arithmetic on dimensions is used, this is because the `fc_arith` package inputs the `calc` package.

The `ansregion` has a formatting string, it prepends `Answer:` to the correct answer. You can change this word using the command `\fmtAnswer` the definition of which is

```
\newcommand{\fmtAnswer}{Answer:}
```

### 3.9. The `\cbOperation` Command

The `\cbOperation` command creates a combo box, named `ChooseOperation`, that lists a menu of arithmetic operations for the user to select.

The syntax for `\cbOperation` is

```
\cbOperation[<eform_parameters>]
```

The default appearance properties are pre-defined as

```
\newcommand{\cBGBorder}{\BC{0 0 0}\BG{.75 .75 .75}
\autoCenter{n}\textSize{0}\textColor{0 0 0}}
```

You can redefine `\cBGBorder`, or you can introduce a few changes through the optional parameter.

The dimensions of `ChooseOperation` field can be set using `\setDimOf`:

```
\setDimOf{ChooseOperation}{1.38in-10bp}{0.24in}
```

The font used in the list box is determined by the value of `\fieldFont`, the default definition is `\newcommand{\fieldFont}{Helv}`.

The strings used in the combo box can be modified by redefining the following commands:

```
\newcommand{\fcAddition}{Addition}
\newcommand{\fcSubtraction}{Subtraction}
\newcommand{\fcMultiplication}{Multiplication}
\newcommand{\fcDivision}{Division}
```

### 3.10. The `\statsFields` Command

The `\statsField` command creates a number of fields that hold the statistics of the user's attempts at answering arithmetic problems.

The following definitions are used.

```
\setDimOf{fcSF}{0.37in}{0.25in}
\newcommand{\statsFieldOpColor}{1 0 0}
\newcommand{\statsFieldColor}{blue}
```

The first is to set the dimensions of each field, they all have the same dimension. The second sets the color for the operations of  $+$ ,  $-$ ,  $\times$  and  $\div$ ; the default color for the labeling of these operations is red ( $\{1\ 0\ 0\}$  in the RGB color space. The `\statsFieldColor` is a color of typeset content and its default is blue. (`\statsFieldOpColor` is used in form fields, while `\statsFieldColor` is  $\LaTeX$ ed.)

## 4. Other Customizations

In this section, we itemize various other customizations not already mentioned.

#### 4.1. Setting the Range and Decimal

The command `\DeclareArithParams`—executed in the preamble only—is used to set the start-up parameters of the **Options** menu [Figure 3](#) on page 6.<sup>4</sup>

The `\DeclareArithParams` takes a series of key-values. We classify the keys by function: setting the intervals and setting the decimal places.

- **Setting the intervals.** Each of these keys takes an interval of the form  $[a, b]$ . Because there is a comma in the interval notation, the interval needs to be enclosed in braces, like so  $\{[a, b]\}$ .
  - **Addition:** `addT={ [a, b] }, addB={ [a, b] }`
  - **Subtraction:** `subT={ [a, b] }, subB={ [a, b] }`
  - **Multiplication:** `mulT={ [a, b] }, mulB={ [a, b] }`
  - **Division:** `divQ={ [a, b] }, divB={ [a, b] }`
- **Setting the decimal places.** The value of these keys takes any of three values none, 1, or 2. In all cases, the default is none.
  - **Addition:** `addDecT=none|1|2, addDecB=none|1|2`
  - **Subtraction:** `subDecT=none|1|2, subDecB=none|1|2`
  - **Multiplication:** `mulDecT=none|1|2, mulDecB=none|1|2`
  - **Division:** `divDecQ=none|1|2, divDecB=none|1|2`

The default definitions are

```
\DeclareArithParams
{%
  addT={ [0, 100] }, addB={ [0, 100] },
  subT={ [0, 100] }, subB={ [0, 100] },
  mulT={ [0, 100] }, mulB={ [0, 10] },
  divQ={ [0, 10] }, divB={ [0, 10] },
}
```

The decimal points are all set to none, by default.

Below is a complete example of setting all parameters.

```
\DeclareArithParams
{%
  addT={ [2, 20] }, addB={ [5, 30] },
  addDecT=none, addDecB=2,
  subT={ [1, 5] }, subB={ [5, 10] },
  subDecT=1, subDecB=2,
  mulT={ [1, 12] }, mulB={ [1, 10] },
  mulDecT=1, mulDecB=2,
  divQ={ [1, 4] }, divB={ [1, 10] },
  divDecQ=1, divDecB=2,
}
```

<sup>4</sup>If the `nomenu` option is taken, the parameters are still used, but user has no way of changing them.

## 4.2. Customization Strings

The following English strings are used in Flash Card. These can all be redefined.

The following is a message that appears in an alert box when your time is up.

```
\newcommand{\timeUpMsg}{Your Time is UP!}
```

These next two are the default messages in the alertbox text field.

```
\newcommand{\rightMsg}{Right!}
\newcommand{\wrongMsg}{Wrong!}
```

The text on the StartAgain and NewProblem.

```
\newcommand{\startAgainMsg}{Start Again}
\newcommand{\newCardMsg}{New Card}
```

The formatting of the ansfield.

```
\newcommand{\fmtAnswer}{Answer:}
```

Labels that appear on the key pad.

```
\newcommand{\kpBack}{Back}
\newcommand{\kpEnter}{Enter}
```

The default setting for the TimeDelay combo box.

```
\newcommand{\fcNoTiming}{No Timing}
```

The listing of arithmetic operations in to combo box titled ChooseOperation.

```
\newcommand{\fcAddition}{Addition}
\newcommand{\fcSubtraction}{Subtraction}
\newcommand{\fcMultiplication}{Multiplication}
\newcommand{\fcDivision}{Division}
```

The text of the menu that, by default, appears in the upper left-corner.

```
\newcommand{\fcSettings}{Settings}
```

Some text strings that are part of the \statsField set.

```
\newcommand{\toggleKeypad}{Toggle Keypad}
\newcommand{\operation}{Operation}
\newcommand{\numCorrect}{Number Correct}
\newcommand{\numAttempted}{Number Attempted}
\newcommand{\percentCorrect}{Percentage Correct}
\newcommand{\timedScores}{Timed Scores}
```

**Customization.** Any change in language strings can be put in the file `fc_custom.def`. Such a file is input, if found, at the end of the package.

## 5. Suggested Layout

The files `fc04.tex` and `fc04_pdf.tex` contain the original layout of PDF Flash Card. We include that layout in this manual for completeness sake.

```

%
% Design your own title
%
\begin{center}
{%
  \LARGE\bfseries\color{blue}PDF Flash Cards\LaTeX
  Elementary Arithmetic
}
}

%
% The arithmetic problem, \arithProb: top, bottom and operation.
% This command is REQUIRED. This command generates three text fields
% stacked so that form a standard arithmetic
%

\arithProb

%
% \alertbox is a text field where a right or wrong message is
% written--REQUIRED
%
% \startAgain clears the statistics field, re-initializes a
% variables--REQUIRED
%
% \inputRegion is where the user enters his/her answer--REQUIRED
%
% \Keypad allows user to enter answer with mouse--OPTIONAL
%
% \newCard random selects a new arithmetic problem (add, sub, mul, div)
% depending on the combo box \cbOperation, described below--REQUIRED
%
% These components can be moved around to a new design, thought I don't
% know what that would be. I have no imagination for design myself.
%
\mbox{\vbox{\smash{\alertbox}\startAgain}\fcSep
  \inputRegion\fcSep\vbox{\smash{\raisebox{4bp}{\Keypad}}\newCard}}

\medskip
%
% \cbTiming is combo box use to set a time limit on answer the
% problem--OPTIONAL
%
% There is also a package option, notimescores, that removes timing
% calculations from the PDF, no statistics, no alerts. The option
% notimescores makes the combo box created by \cbTiming into a readonly

```



```
% field.
%
% \ansField is the field the user enters his/her answer into---REQUIRED
%
% \cbOperation a combo box to select what operation to use---REQUIRED
%
\mbox{\cbTiming\fcSep\ansField\fcSep\cbOperation}

\medskip

%
% This calculation computes the width of the previous row of fields, and
% sets \fcWidth, a dimension in this package, to that width.
%
\settowidth{\fcWidth}{\cbTiming\fcSep\ansField\fcSep\cbOperation}
%
% \statsFields is a collection of text fields to display user
% statistics---OPTIONAL
%
\makebox[\fcWidth][s]{\statsFields}

\end{center}
```