# Readme Simplex Method

FIRST: sorry for my english!

2nd: i'm sorry, i don't know why, but if i try to print a pdf docs from this one, pdf haven't the equations, so i export it in others popular and free formats: .odt (that is more free than .pdf), and html (that is more popular than pdf).

## License

**Gnu free documentation license**.

## Version 0.3

### Input required

A tableau matrix and a base.

If you have the problem of linear programming in standard canonical form:

where:

The **tableau matrix** is:

The base is:

An ordered set of index that imply an inner identity matrix of tableau.

### Output

The program don't do smart things (at least in this version), it do some checks for canonical form and after these checks it try to obtain optimal tableau from input. Optimal tableau is the first tableau with even if there are others solutions (for example in a unlimited ).

Moreover it put on stack the partial results as tableau.

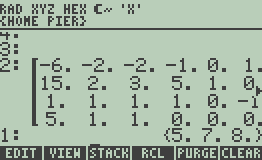
After it try to obtain others optimal tableaus using , but it can find all solutions (if there are more than two ) see the code for details.

Then it put on stack the others optimal tableaus (if any) and a matrix where each row is an optimal canonical base.

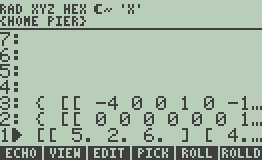
So you can use it only for boring operations, not for "smart" steps.

### Example

Inputs (tableau & base)



Outputs



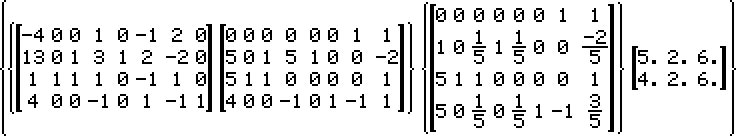
(this is an edit view)

On 3rd level, there are partial tableaus and the first optimal tableau (as list)

On 2nd level there are the others optimal tableaus (as list)

On 1st level there is the matrix where each row of it is an optimal canonical base.

All 3 levels are show below as list (the program don't do it automatically)



As you can see, the optimal tableau (2nd matrix in the first sublist) allow a lot of others optimal tableaus, using . The program, because isn't simple (mainly for computation cost) at least for me, can't find all optimal tableaus if there are many zeroes.

The next versions focus on efficiency and some others point of effectiveness (the best is: make a C ARM code, so it will be ~1000x faster on hp50g but anyway it will be much less portable because userRPL runs on emulators and others calulators).

## Version 0.1

### Input required

A tableau matrix and a base.

If you have the problem of linear programming in standard canonical form:

where:

The **tableau matrix** is:

The base is:

An ordered set of index that imply an inner identity matrix of tableau.

### Output

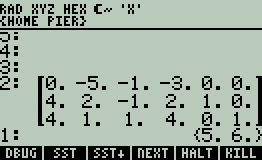
The program don't do smart things (at least in this version), it do some checks for canonical form and after these checks it try to obtain optimal tableau from input. Optimal tableau is the first tableau with even if there are others solutions (for example in a unlimited ).

Moreover it put on stack the partial results as tableau.

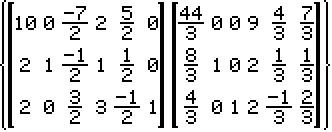
So you can use it only for boring operations, not for "smart" steps.

### Example

Inputs (tableau & base)



Outputs



First is the partial tableau, last is the optimal tableau.