Trabalho Computacional Opcional

Data da entrega: 27 de junho, até 22h, por email: luziane@dcc.ufrj.br

*** Obs: Trabalhos entregues depois desse prazo serão desconsiderados. ***

1 Entrada de Dados

Neste trabalho é esperado que o aluno desenvolva um programa que resolva problemas de programação linear (PPL) com n variáveis originais e m restrições de igualdade e/ou desigualdade, utilizando o método simplex em formato tabular.

O programa deve ser feito **exclusivamente** em uma das 3 linguagens abaixo:

- Python 2. Obs: Python 3 não será aceito.
- C
- C++

Os problemas a serem resolvidos por esse programa podem ter estruturas variadas:

- ser de minimização ou maximização;
- cada restrição pode ser de igualdade ou desigualdade (neste segundo caso, podendo ser do tipo "$\geq$" ou "$\leq$");
- cada variável x_i pode ser ≥ 0, ≤ 0 ou irrestrita.

Portanto, é muito importante que a entrada contemple os dados do problema, detalhando não só os valores dos coeficientes e limites envolvidos, como também o tipo de cada uma das restrições e variáveis, e se o mesmo é de minimização ou maximização.

Essa **entrada** de informações deve ser feita **via teclado do computador**, após a chamada do arquivo executável do programa; a primeira linha a ser digitada possui um número inteiro p, sem espaço prévio, indicando a quantidade de problemas a serem testados **em uma única chamada** do arquivo executável.

Da segunda linha em diante entrarão as informações relativas aos p problemas, formando p "blocos" de linhas. **Cada bloco** de linhas será da seguinte forma:

- Primeira linha do bloco:
  São dois números inteiros (n e m), separados por espaço; o primeiro número (n) corresponde ao número de variáveis e o segundo número (m) ao número de restrições que o problema em questão possui em seu formato original.
- Segunda linha do bloco:
  Inicia com a *string* **min** se for um problema de minimização ou **max** se for um problema de maximização. Em seguida, na mesma linha e separados por um espaço, são digitados n números reais (todos com 4 casas decimais) que correspondem aos coeficientes de cada variável na função objetivo. Lembre-se que, caso uma variável não apareça na função objetivo, o coeficiente atribuído a ela deve ser 0.0000

- Próximas linhas do bloco:
  Serão digitadas m linhas, onde cada uma delas corresponde a uma restrição. Portanto, em cada linha, são digitados n números reais (todos com 4 casas decimais, separados por um espaço) que correspondem aos coeficientes de cada variável na restrição; em seguida, na mesma linha e precedido por um espaço, será digitado o símbolo “>” ou “<” ou “=” se a restrição for do tipo “maior ou igual”, “menor ou igual” ou “de igualdade”, respectivamente. Por fim, precedido por um espaço, a linha se encerra com um valor real (com 4 casas decimais) que está do lado direito da restrição.
  Obs: Lembre-se que, caso uma variável não apareça em alguma restrição, o coeficiente atribuído a ela deve ser 0.0000

- Última linha do bloco:
  Serão digitadas n strings, separadas por um espaço, onde cada uma delas corresponde à “restrição de sinal” de cada variável. Portanto, será digitado o símbolo “>” ou “<” ou “L” se a variável for do tipo “maior ou igual a zero”, “menor ou igual a zero” ou “livre de sinal”, respectivamente, para cada variável.

  Valores máximos:

- Cada problema testado terá no máximo 50 variáveis originais ($n \in \{1, 2, \ldots, 50\}$);

- Cada problema testado terá no máximo 100 restrições originais ($m \in \{1, 2, \ldots, 100\}$).

1.1 Exemplo

Para compreender melhor como é a forma de entrada dos dados no terminal, considere o exemplo a seguir.

Serão repassadas para o programa as informações sobre os 5 PPL's abaixo:

- **Primeiro problema - P1:**
$$\begin{array}{rrcrcr}
\min & 50x_1 & + & 100x_2 & & \\
\text{s.a} & 7x_1 & + & 2x_2 & \geq & 28 \\
 & 2x_1 & + & 12x_2 & \geq & 24 \\
 & 7x_1 & - & 2x_2 & \geq & 14 \\
 & x_1, & & x_2 & \geq & 0
\end{array}$$

- **Segundo problema - P2:**
$$\begin{array}{rrcrcr}
\min & 7x_1 & - & 2x_2 & & \\
\text{s.a} & 7x_1 & - & 2x_2 & \geq & 28 \\
 & 2x_1 & - & 12x_2 & \geq & 24 \\
 & 7x_1 & + & 2x_2 & \geq & 14 \\
 & x_1 & \geq 0, & x_2 & \leq & 0
\end{array}$$

- **Terceiro problema - P3:**
$$\begin{array}{rrcrcr}
\min & 7x_1 & - & 2x_2 & & \\
\text{s.a} & 7x_1 & + & 2x_2 & \geq & 28 \\
 & 2x_1 & + & 12x_2 & \geq & 24 \\
 & 7x_1 & - & 2x_2 & \geq & 14 \\
 & x_1, & & x_2 & \geq & 0
\end{array}$$

- **Quarto problema - P4:**

$$\begin{array}{rrcrcr} \max & x_1 & + & x_2 & & \\ \text{s.a} & 7x_1 & + & 2x_2 & \geq & 28 \\ & 2x_1 & + & 12x_2 & \geq & 24 \\ & 7x_1 & - & 2x_2 & \geq & 14 \\ & x_1, & & x_2 & \geq & 0 \end{array}$$

- **Quinto problema - P5:**

$$\begin{array}{rrcrcl} \max & 2x_1 & + & 4x_2 & & \\ \text{s.a} & 7x_1 & + & 2x_2 & \geq & 28 \\ & 2x_1 & + & 12x_2 & \leq & 24 \\ & 7x_1 & - & 2x_2 & = & 14 \\ & x_1 & \geq 0, & x_2 & & \text{irrestrito} \end{array}$$

Seguindo as instruções expostas, caso desejemos resolver esses 5 problemas de uma vez (em uma única chamada do programa), a entrada de dados a ser digitada no terminal será a seguir:

```
5
2 3
min 50.0000 100.0000
7.0000 2.0000 > 28.0000
2.0000 12.0000 > 24.0000
7.0000 -2.0000 > 14.0000
> >
2 3
min 7.0000 -2.0000
7.0000 -2.0000 > 28.0000
2.0000 -12.0000 > 24.0000
7.0000 2.0000 > 14.0000
> <
2 3
min 7.0000 -2.0000
7.0000 2.0000 > 28.0000
2.0000 12.0000 > 24.0000
7.0000 -2.0000 > 14.0000
> >
2 3
max 1.0000 1.0000
7.0000 2.0000 > 28.0000
2.0000 12.0000 > 24.0000
7.0000 -2.0000 > 14.0000
> >
2 3
max 2.0000 4.0000
7.0000 2.0000 > 28.0000
2.0000 12.0000 < 24.0000
7.0000 -2.0000 = 14.0000
> L
```

2 Recomendações

2.1 Sobre a padronização

Ao tentar padronizar os problemas recebidos para colocar no formato de igualdade, muitos ajustes podem ser necessários. É muito importante que as alterações (necessárias para colocar no formato desejado para executar o simplex) sejam realizadas na ordem indicada abaixo; essa ordem dos passos de conversão influencia na posição em que as colunas aparecerão nas tabelas, o que é de grande relevância quando o algoritmo for validado utilizando problemas com muitas variáveis.

- **Primeira alteração:** Substitua cada variável que seja do tipo ≤ 0, colocando a variável nova na "posição" da antiga. Por exemplo, caso $x_3 \leq 0$, a nova variável $x_3' = -x_3$ deve substituir a antiga; os valores dos coeficientes de x_3' na matriz A e na função objetivo devem ocupar o lugar dos coeficientes que antes pertenciam a x_3.
- **Segunda alteração:** Substitua cada variável que seja irrestrita por outras duas variáveis novas, ocupando a "posição" da antiga. Por exemplo, caso a variável x_3 seja irrestrita, os valores dos coeficientes das novas variáveis na matriz A e na função objetivo devem ocupar o lugar dos coeficientes que antes pertenciam a x_3 (como são duas novas variáveis, ocuparão a terceira e quarta posições).
- **Terceira alteração:** Como as restrições podem ser de todos os tipos, caso necessário, o programa deve tentar padronizar colocando todas as restrições no formato de igualdade; para cada restrição de desigualdade, seguindo da primeira restrição em direção à última (nessa ordem), some ou subtraia as variáveis de folga necessárias.
- **Quarta alteração:** como o problema pode ser de minimização ou de maximização, caso necesário, converta-o para minimização.

Dado que as restrições iniciais eram mistas, a implementação do simplex deve ser feita utilizando o método das duas fases. Além de facilitar encontrar uma base inicial para a execução do simplex, a fase I também ajudará a identificar o caso de PPL inviável.

Para iniciar a fase I, apenas para cada restrição sem variável de folga ou com variável de folga subtraída, seguindo da primeira restrição em direção à última (nessa ordem), introduza uma variável artificial. Somente após esse procedimento você deve montar a tabela inicial do método simplex para a fase I, utilizando a função objetivo adequada para essa fase.

Ao final da fase I, identifique se o PPL original é inviável ou não. Caso seja viável, gere a primeira tabela da fase II a partir da última tabela da fase I, retirando as variáveis artificiais e corrigindo a linha z.

2.1.1 Exemplo

Considere como exemplo para o procedimento de padronização o quinto PPL (P5). Seguindo os passos para a padronização, a primeira coisa a ser feita é a substituição de variáveis que sejam ≤ 0.

$$\begin{array}{rrcrcl}
\max & 2x_1 & + & 4x_2 & & \\
\text{s.a} & 7x_1 & + & 2x_2 & \geq & 28 \\
 & 2x_1 & + & 12x_2 & \leq & 24 \\
 & 7x_1 & - & 2x_2 & = & 14 \\
 & x_1 & \geq 0, & x_2 & & \text{irrestrito}
\end{array}$$

Como P5 não possui esse tipo de variável, vamos para o passo seguinte.

A segunda coisa a ser feita é a substituição de variáveis que sejam irrestritas. A variável x_2 deverá ser substituída por duas outras ($x_2 = x_2' - x_2''$) cujo posicionamento é o mesmo que o da antiga x_2 (elas ficarão logo após x_1, tal qual x_2 ficava). Logo, sua formulação passa a ser:

$$\begin{array}{rrcrcrcl}
\max & 2x_1 & + & 4x_2' & - & 4x_2'' & & \\
\text{s.a} & 7x_1 & + & 2x_2' & - & 2x_2'' & \geq & 28 \\
 & 2x_1 & + & 12x_2' & - & 12x_2'' & \leq & 24 \\
 & 7x_1 & - & 2x_2' & + & 2x_2'' & = & 14 \\
 & x_1, & & x_2', & & x_2'' & \geq & 0
\end{array}$$

O terceiro passo é a introdução das variáveis de folga, da primeira até a última restrição, quando necessário:

$$\begin{array}{rrcrcrcrcrcl}
\max & 2x_1 & + & 4x_2' & - & 4x_2'' & & & & & & \\
\text{s.a} & 7x_1 & + & 2x_2' & - & 2x_2'' & - & x_3 & & & = & 28 \\
 & 2x_1 & + & 12x_2' & - & 12x_2'' & & & + & x_4 & = & 24 \\
 & 7x_1 & - & 2x_2' & + & 2x_2'' & & & & & = & 14 \\
 & x_1, & & x_2', & & x_2'', & & x_3, & & x_4 & \geq & 0
\end{array}$$

Por fim, convertendo o problema para minimização, encerramos o processo de padronização:

$$\begin{array}{rrcrcrcrcrcl}
\min & -2x_1 & - & 4x_2' & + & 4x_2'' & & & & & & \\
\text{s.a} & 7x_1 & + & 2x_2' & - & 2x_2'' & - & x_3 & & & = & 28 \\
 & 2x_1 & + & 12x_2' & - & 12x_2'' & & & + & x_4 & = & 24 \\
 & 7x_1 & - & 2x_2' & + & 2x_2'' & & & & & = & 14 \\
 & x_1, & & x_2', & & x_2'', & & x_3, & & x_4 & \geq & 0
\end{array}$$

Para iniciar a fase I, vamos introduzir as variáveis artificiais necessárias x_5 e x_6 (apenas para cada restrição sem variável de folga ou com variável de folga subtraída, seguindo da primeira restrição em direção à última).

$$\begin{array}{rrcrcrcrcrcrcrcl}
\min & & & & & & & & & & & x_5 & + & x_6 & & \\
\text{s.a} & 7x_1 & + & 2x_2' & - & 2x_2'' & - & x_3 & & & + & x_5 & & & = & 28 \\
 & 2x_1 & + & 12x_2' & - & 12x_2'' & & & + & x_4 & & & & & = & 24 \\
 & 7x_1 & - & 2x_2' & + & 2x_2'' & & & & & & & + & x_6 & = & 14 \\
 & x_1, & & x_2', & & x_2'', & & x_3, & & x_4, & & x_5, & & x_6 & \geq & 0
\end{array}$$

2.2 Observações

Observe que é possível utilizar rotinas ou pacotes já prontos que lhe permitam fazer a leitura de uma matriz, inverter matrizes, etc., mas não serão aceitos trabalhos em que o código dos métodos seja uma rotina pré-pronta (ou seja, não devem ser utilizados códigos com o método já completamente/parcialmente implementado ou códigos copiados de amigos ou da internet). Se isso for identificado, o trabalho terá sua nota completamente zerada, independente de quem realmente fez a implementação ou de quem apenas a copiou. Ou seja, é importante que os métodos que você vai usar para resolver o que foi pedido seja implementado por você e não por outra pessoa.

Plágio (de um colega de sala, de programas já prontos na internet, ou de qualquer outra fonte) é uma violação de conduta acadêmica, cuja punição imedita é a nota zerada deste trabalho, podendo ter punição mais severa dependendo do caso (abertura de processo disciplinar).

Não imprima nada além do que foi solicidado no terminal, mesmo que sejam **frases** como "entre com o PPL" ou "a solução encontrada foi:". Os números e dados do PPL devem ser inseridos na ordem como mencionado neste trabalho, assim como os dados de saída, respeitando o que deve ser colocado em cada linha, o espaço entre os números e a quantidade de casas decimais dos números reais.

Todas as rotinas, funções, pedaços de código, código principal, ..., devem ser colocados em um **único arquivo**, cujo nome deve ser "DRE.py" ou "DRE.c" ou "DRE.cpp". Por exemplo, se eu sou aluna de PL com DRE 099237121 e desejo entregar o trabalho, o meu arquivo **único** deve se chamar "099237121.py" ou "099237121.c" ou "099237121.cpp".

Como a entrada e a saída de dados ocorre no terminal, para que não haja dúvidas, relato a seguir o que deve ocorrer: após iniciada a execucação do programa, deve ser digitado o valor de p. Logo em seguida, devem ser digitados os dados sobre o primeiro problema; após resolução, será impresso no terminal o resultado do problema conforme descrito na seção a seguir. Após o término da resolução desse primeiro problema, o programa aguarda a entrada dos dados no terminal do segundo problema; após a resolução, será impresso no terminal o resultado do mesmo conforme descrito na seção a seguir. E assim sucessivamente, até que sejam digitados os dados sobre o p-ésmo problema e impresso no terminal o seu resultado, o que encerra a execução do programa.

3 Saída de Dados

Após a leitura de cada um dos p problemas, o resultado dele deve ser impresso no terminal, formando ao final da execução do programa p blocos de saída. Cada bloco **deve ser precedido** por uma linha com três tracinhos impressos: `- - -` (três tracinhos separados entre si por um espaço)

Cada bloco deve exibir os dados na saída conforme descrito a seguir.

- A primeira coisa a ser impressa em cada bloco deve ser a tabela inicial da fase I; a formatação dessa tabela (e de qualquer outra do método simplex) é descrita na próxima subseção.
- A segunda coisa a ser impressa em cada bloco deve ser a tabela inicial da fase II (caso o PPL não seja inviável); a formatação dessa tabela (e de qualquer outra do método simplex) é descrita na próxima subseção.
- Primeira linha subsequente do bloco:
  Deve ser impresso um número inteiro t, dependendo do tipo de solução encontrada:

  1 se solução única;

  2 se múltiplas soluções, que residem em uma lateral finita da região viável;

  3 se múltiplas soluções, que residem em uma lateral infinita da região viável;

  4 se solução ilimitada;

  5 se o problema é sem solução (região viável vazia).
- Apenas se t for 1 ou 2, o aluno deve imprimir todos os vértices ótimos, um por linha, quantos forem. Para tanto, em cada linha correspondente a um vértice, será impresso a letra "V" seguida de um número inteiro (sem espaço) que indica a ordem de impressão do vértice em questão. Por fim, na mesma linha e precedidos por um espaço, são impressos n números reais com 4 casas decimais, separados por 1 espaço, onde cada número é o valor de uma variável original no vértice ótimo.

- Apenas se t for 3, o aluno deve imprimir todos os vértices ótimos, um por linha, quantos forem. Para tanto, em cada linha correspondente a um vértice, será impresso a letra "V" seguida de um número inteiro (sem espaço) que indica a ordem de impressão do vértice em questão. Por fim, na mesma linha e precedidos por um espaço, são impressos n números reais com 4 casas decimais, separados por 1 espaço, onde cada número é o valor de uma variável original no vértice ótimo.

  Após a impressão de todos os vértices ótimos, o programa deve imprimir nas linhas seguintes as direções extremas da lateral ótima (quantas forem), indicando a partir de qual vértice essa direção deve ser considerada. Portanto, cada direção extrema ótima deve ser impressa, uma por linha, como uma sequência de n números reais com 4 casas decimais correspondendo a cada uma das componentes da direção extrema. A linha deve iniciar com a letra "D" e um número inteiro (sem espaço entre eles) indicando o vértice a partir do qual a direção extrema deve ser considerada.

  Ou seja, primeiro são impressos todos os vértices ótimos e depois todas as direções ótimas.

- Para finalizar o bloco: se t for 1, 2 ou 3, a última linha deve conter um número real com 4 casas decimais que é o valor da imagem ótima do PPL original (lembre-se de observar se houve troca de max para min); se t for 4, nessa linha deve ser impresso "+ inf " ou "- inf" (dependendo se o problema original for maximização ou minimização, respectivamente); se t for 5, nessa linha deve ser impresso "imp".

Observação: após o último bloco (bloco referente ao $p-$ésimo problema) deve ser impressa uma linha com três tracinhos: `- - -` (três tracinhos separados entre si por um expaço).

3.1 Como imprimir uma tabela do Simplex

Considere o problema P5 após todos os ajustes para ficar na formatação desejada para iniciar a fase I do método das duas fases (veja Seção 2.1.1). A tabela escrita "à mão" (em sala) seria a seguinte:

	x_1	x'_2	x''_2	x_3	x_4	x_5	x_6	RHS
z	-14	0	0	1	0	0	0	-42
x_5	7	2	-2	-1	0	1	0	28
x_4	2	12	-12	0	1	0	0	24
x_6	7	-2	2	0	0	0	1	14

A impressão de cada tabela necessária na saída deve ter apenas as colunas com números, escritos com 4 casas decimais. Para o exemplo acima, a tabela a ser impressa na saída corresponde a uma sequência de 4 linhas, cada uma delas com 8 números reais (com 4 casas decimais) separados por um espaço:

```
-14.0000 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000 -42.0000
7.0000 2.0000 -2.0000 -1.0000 0.0000 1.0000 0.0000 28.0000
2.0000 12.0000 -12.0000 0.0000 1.0000 0.0000 0.0000 24.0000
7.0000 -2.0000 2.0000 0.0000 0.0000 0.0000 1.0000 14.0000
```

3.2 Exemplo

Para compreender melhor como é a forma de saída dos dados no terminal, considere os problemas apresentados na Seção 1.1.

Após a resolução, as soluções encontradas são:

- **Primeiro problema - P1:** Solução ótima única $x^* = (3.6;\ 1.4)^T$ com imagem $z^* = 320$.
- **Segundo problema - P2:** Solução ótima alternativa (ou múltiplas), formada por todo o segmento de reta cujos extremos são $E_1 = (3.6;\ 1.4)^T$ e $E_2 = (3;\ 3.5)^T$. Ou seja $x^* = \alpha_1 E_1 + \alpha_2 E_2$, com $\alpha_1 + \alpha_2 = 1$ e $\alpha_1,\ \alpha_2 \geq 0$.
  Lembre-se que aqui me refiro ao problema original, cuja segunda variável era da forma $x_2 \leq 0$.
- **Terceiro problema - P3:** Solução ótima alternativa (ou múltiplas), formada pela semi-reta iniciando no ponto extremo $E_1 = (3;\ 3.5)^T$ e direção $d_1 = (0.0714286;\ 0.25)^T$ (múltipla de $(1;\ 3.5)^T$).
- **Quarto problema - P4:** Solução ilimitada, com $Z^* \to +\infty$ (lembre-se que aqui me refiro à função original, que foi maximizada).
- **Quinto problema - P5:** O problema é sem solução, com região viável vazia.

Portanto, a saída esperada do programa quando os 5 problemas do exemplo são resolvidos em uma única execução é:

```
- - -
-16.0000 -12.0000 1.0000 1.0000 1.0000 0.0000 0.0000 0.0000 -66.0000
7.0000 2.0000 -1.0000 0.0000 0.0000 1.0000 0.0000 0.0000 28.0000
2.0000 12.0000 0.0000 -1.0000 0.0000 0.0000 1.0000 0.0000 24.0000
7.0000 -2.0000 0.0000 0.0000 -1.0000 0.0000 0.0000 1.0000 14.0000
0.0000 0.0000 5.0000 7.5000 0.0000 -320.0000
-0.0000 0.0000 -1.1000 0.3500 1.0000 8.4000
0.0000 1.0000 0.0250 -0.0875 0.0000 1.4000
1.0000 0.0000 -0.1500 0.0250 0.0000 3.6000
1
V1 3.6000 1.4000
320.0000
- - -
-16.0000 -12.0000 1.0000 1.0000 1.0000 0.0000 0.0000 0.0000 -66.0000
7.0000 2.0000 -1.0000 0.0000 0.0000 1.0000 0.0000 0.0000 28.0000
2.0000 12.0000 0.0000 -1.0000 0.0000 0.0000 1.0000 0.0000 24.0000
7.0000 -2.0000 0.0000 0.0000 -1.0000 0.0000 0.0000 1.0000 14.0000
0.0000 0.0000 1.0000 0.0000 0.0000 -28.0000
-0.0000 0.0000 -1.1000 0.3500 1.0000 8.4000
0.0000 1.0000 0.0250 -0.0875 0.0000 1.4000
1.0000 0.0000 -0.1500 0.0250 0.0000 3.6000
2
V1 3.6000 1.4000
V2 3.0000 3.5000
28.0000
- - -
-16.0000 -12.0000 1.0000 1.0000 1.0000 0.0000 0.0000 0.0000 -66.0000
7.0000 2.0000 -1.0000 0.0000 0.0000 1.0000 0.0000 0.0000 28.0000
2.0000 12.0000 0.0000 -1.0000 0.0000 0.0000 1.0000 0.0000 24.0000
7.0000 -2.0000 0.0000 0.0000 -1.0000 0.0000 0.0000 1.0000 14.0000
0.0000 0.0000 1.1000 -0.3500 0.0000 -22.4000
-0.0000 0.0000 -1.1000 0.3500 1.0000 8.4000
0.0000 1.0000 0.0250 -0.0875 0.0000 1.4000
1.0000 0.0000 -0.1500 0.0250 0.0000 3.6000
3
V1 3.0000 3.5000
D1 0.0714 0.2500
14.0000
- - -
-16.0000 -12.0000 1.0000 1.0000 1.0000 0.0000 0.0000 0.0000 -66.0000
7.0000 2.0000 -1.0000 0.0000 0.0000 1.0000 0.0000 0.0000 28.0000
2.0000 12.0000 0.0000 1.0000 0.0000 0.0000 1.0000 0.0000 24.0000
7.0000 -2.0000 0.0000 0.0000 1.0000 0.0000 0.0000 1.0000 14.0000
0.0000 0.0000 -0.1250 -0.0625 0.0000  5.0000
0.0000 0.0000 -1.1000  0.3500 1.0000  8.4000
0.0000 1.0000  0.0250 -0.0875 0.0000  1.4000
1.0000 0.0000 -0.1500  0.0250 0.0000  3.6000
4
+ inf
- - -
-14.0000 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000 -42.0000
7.0000 2.0000 -2.0000 -1.0000 0.0000 1.0000 0.0000 28.0000
2.0000 12.0000 -12.0000 0.0000 1.0000 0.0000 0.0000 24.0000
7.0000 -2.0000 2.0000 0.0000 0.0000 0.0000 1.0000 14.0000
5
imp
- - -
```