

5COP093 - Trabalho T2

Utilize as ferramentas **flex** e **bison** para criar um programa para manipulação matemática chamado **dcmat**. Neste programa as funções podem ser formadas por:

operadores binários: + - * / ^ %

operadores unários: + -

delimitadores: ()

funções: sen cos tan abs

números inteiros

números reais

variável x

Números reais devem ser formados pela seguinte expressão regular:

$$[0-9]^+ \cdot [0-9]^+$$

Exemplos de expressões válidas são:

sen(x)

-x*sen(x)

-10*45^2

(1+3)*78

17.78+3

-x*sen(x)+cos(x)*(tan(x+3.78)/4.2)

Crie a sua gramática de forma que as devidas precedências sejam respeitadas. O programa quando iniciado apresenta um **prompt** de comando através do símbolo:

>

Ao se pressionar a tecla **ENTER** em uma linha sem nenhum comando, uma nova linha do prompt de comando com o símbolo > deve ser criada, sem que nenhum erro seja gerado. Neste **prompt** devem ser digitados os comandos, os quais são finalizados através do símbolo de ponto-e-vírgula, com exceção do comando **quit**. O **dcmat** fica aguardando comandos até que o comando **quit** seja digitado.

Os comandos e sua sintaxe são detalhados a seguir.

1 Comandos do dcmat

1.1 show settings

Sintaxe: `show settings;`

Este comando mostra o conteúdo das variáveis internas do programa. Exemplo de utilização:

```
>show settings;

h_view_lo: -6.500000
h_view_hi: 6.500000
v_view_lo: -3.500000
v_view_hi: 3.500000
integral_steps: 1000
```

```
Draw Axis: ON.
Connect Dots: OFF.
```

```
>
```

1.2 reset settings

Sintaxe: `reset settings;`

Este comando restaura os valores padrão das variáveis internas do programa. Exemplo de utilização:

```
>reset settings;
>
```

1.3 quit

Sintaxe: `quit`

Sai do programa dcmat. Exemplo de utilização:

```
>quit
```

1.4 set h_view

Sintaxe: `set h_view [valor float] : [valor float] ;`

Este comando seta os parâmetros de visualização horizontal do eixo X para a geração de gráficos de funções. Exemplo de utilização:

```
>set h_view -10.5:7.2;
```

1.5 set v_view

Sintaxe: `set v_view [valor float] : [valor float] ;`

Este comando seta os parâmetros de visualização vertical do eixo Y para a geração de gráficos de funções. Exemplo de utilização:

```
>set v_view -5:+10.2;
```

1.6 set axis on

Sintaxe: `set axis on;`

Este comando faz com que os eixos X e Y sejam desenhados ao se plotar um gráfico de função. Exemplo de utilização:

```
>set axis on;  
>
```

1.7 set axis off

Sintaxe: `set axis off;`

Este comando faz com que os eixos X e Y **NÃO** sejam desenhados ao se plotar um gráfico de função. Exemplo de utilização:

```
>set axis off;  
>
```

1.8 plot

Sintaxe: `plot;`

Este comando plota a última função que foi inserida. No caso de nenhuma função haver sido inserida ainda, o comando mostra a mensagem: `No function defined!` Exemplo de utilização quando nenhuma função foi inserida:

```
>plot;
```

```
No Function defined!
```

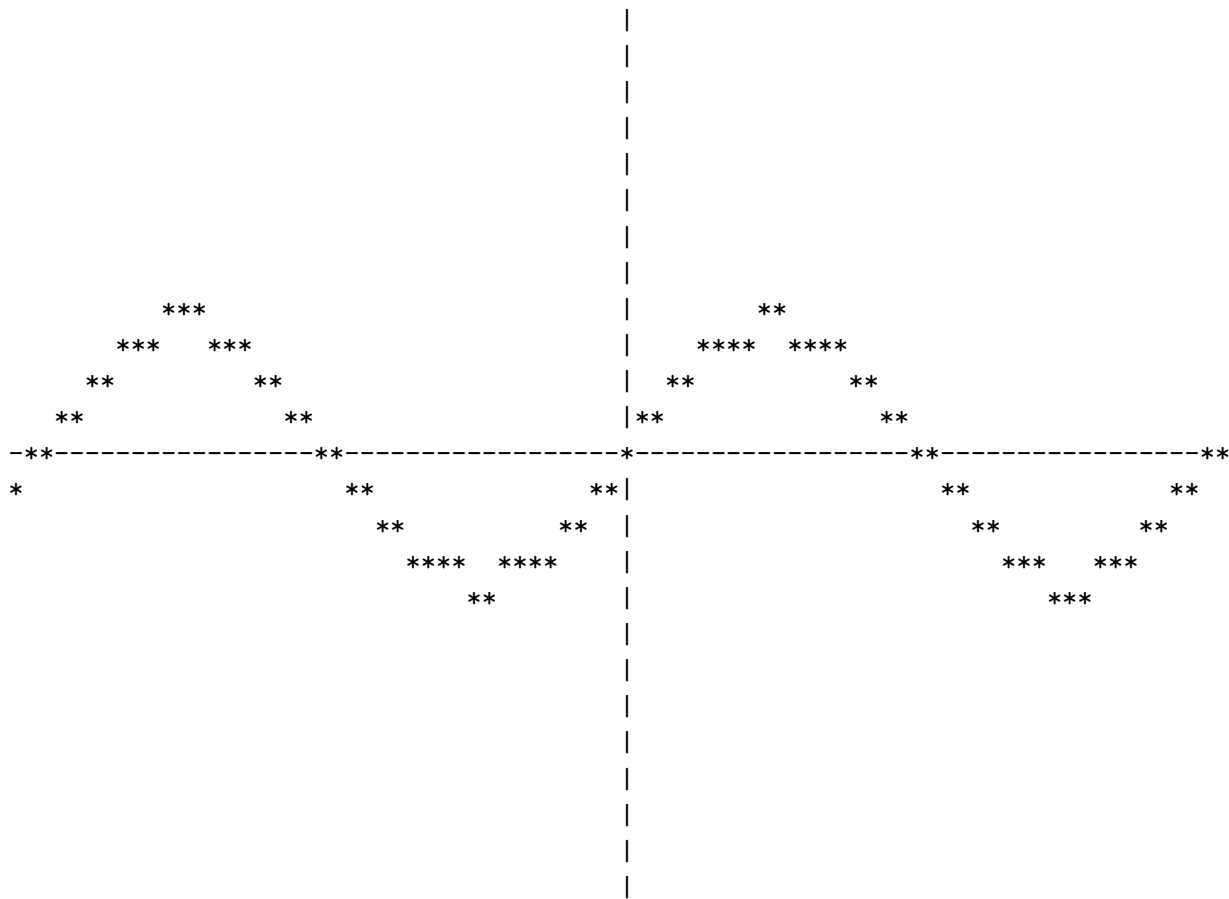
```
>
```

1.9 plot

Sintaxe: `plot([função]);`

Este comando é utilizado para se plotar funções matemáticas. Exemplo de utilização do comando `plot`, quando os valores padrão de plotagem horizontal e vertical são utilizados e com o desenho dos eixos X e Y habilitado:

```
>plot(sen(x));
```

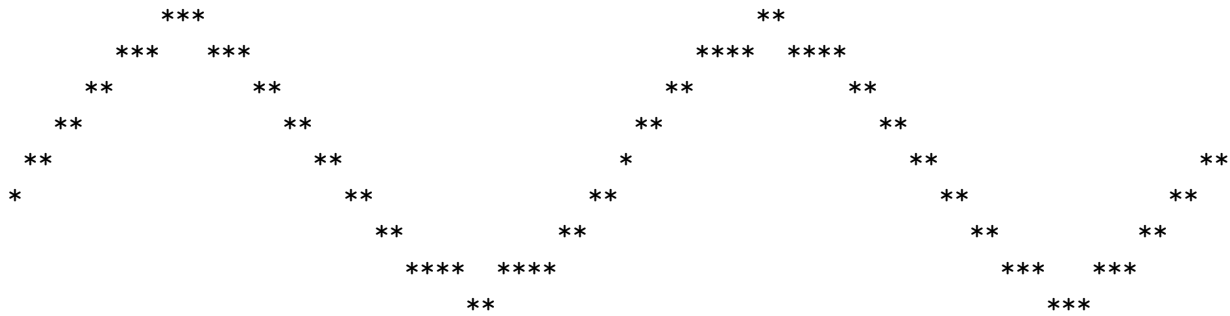


```
>
```

O programa plota gráficos com resolução padrão de 25 linhas por 80 colunas em modo texto, sendo que o caractere `*` é utilizado como *pixel*.

Exemplo de utilização do comando `plot`, quando os valores padrão de plotagem horizontal e vertical são utilizados e com o desenho dos eixos X e Y **desabilitado**:

```
>set axis off;  
>plot(sen(x));
```



>

1.10 Impressão em RPN

Caso uma função seja diretamente introduzida, sua forma RPN é mostrada pelo programa. Exemplo:

```
>sen(x)*(1+x)
```

Function in RPN format:

```
x SEN 1.000000 x + *
```

```
>
```

Observe que a função é inserida sem a utilização de ponto-e-vírgula.

IMPORTANTE: A função introduzida para ser impressa na forma RPN não é armazenada e não altera a função previamente armazenada, caso exista, pelo comando `plot`.

1.11 set integral steps

Sintaxe: `set integral_steps [valor inteiro]` ;

Este comando ajusta a quantidade de passos a serem utilizados ao se calcular o valor numérico de uma integral utilizando-se a *Soma de Riemann*. Exemplo de utilização:

```
>set integral_steps 350;
```

```
>
```

1.12 integrate

Sintaxe: `integrate([limite inferior] : [limite superior] , [função])` ;

Este comando é utilizado para se calcular o valor numérico de uma integral entre o intervalo definido pelos valores *limite inferior* e *limite superior* para uma determinada função. Exemplo de utilização:

```
>integrate(0:6.28,sen(x));
```

```
0.000010
```

```
>
```

No exemplo apresentado é calculado o valor numérico da integral:

$$\int_0^{6.28} \text{sen}(x) \, dx = -\cos(x) \Big|_0^{6.28}$$

1.13 matrix

Sintaxe: `matrix = [[valor {, valor}*] {,[valor {, valor}*] }*]`;

* Indica uma repetição de zero ou mais vezes do termo que este entre { e }.
valor indica um número real.

Este comando é utilizado para se realizar a entrada de uma matriz no sistema. Exemplo de utilização:

```
>matrix = [[1]];  
>
```

No exemplo apresentado, foi inserida a matriz:

$$\begin{bmatrix} 1 \end{bmatrix}$$

Outro exemplo de utilização:

```
>matrix = [ [1,2], [3,4,5], [6] ];  
>
```

No exemplo apresentado, foi inserida a matriz:

$$\begin{bmatrix} 1 & 2 & 0 \\ 3 & 4 & 5 \\ 6 & 0 & 0 \end{bmatrix}$$

Observe que o número de colunas será determinado pela linha que contiver o maior número de elementos, sendo que as demais linhas terão o seu tamanho completado inserindo-se valores iguais a zero.

1.14 show matrix

Sintaxe: `show matrix`;

Este comando mostra, em notação científica, a matriz que foi previamente inserida no sistema. Exemplo de utilização:

```
>show matrix;
```

```
+--                                +-  
|  1.000000e+00  2.000000e+00  0.000000e+00 |  
|  3.000000e+00  4.000000e+00  5.000000e+00 |  
|  6.000000e+00  0.000000e+00  0.000000e+00 |  
+---                                +-  
  
>
```

Considere que a seguinte matriz foi inserida:

$$\begin{bmatrix} -1 & 2 \\ 3 & -4 \end{bmatrix}$$

Observe que a matriz possui números negativos. Se a matriz for impressa, os números devem ser impressos de forma alinhada. Exemplo:

```
>matrix = [[-1,2],[3,-4]];
```

```
>show matrix;
```

```
+--                               +-
|  -1.000000e+00  2.000000e+00  |
|   3.000000e+00 -4.000000e+00  |
+---                               +-
```

```
>
```

Se não houver nenhuma matriz inserida, o comando mostra a mensagem: `No Matrix defined!` Exemplo de utilização quando nenhuma matriz foi inserida:

```
>show matrix;
```

```
No Matrix defined!
```

```
>
```

1.15 solve determinant

Sintaxe: `solve determinant;`

Este comando calcula o determinante de uma matriz quadrada n por n .

Suponha que se deseje computar o determinante da seguinte matriz:

$$\begin{bmatrix} 10 & 2 & 1 \\ 1 & 5 & 1 \\ 2 & 3 & 10 \end{bmatrix}$$

A computação do determinante da matriz apresentada seria da seguinte forma:

```
>matrix = [[10,2,1],[1,5,1],[2,3,10]];
```

```
>solve determinant;
```

```
447.000000
```

```
>
```

Se não houver nenhuma matriz inserida, o comando mostra a mensagem: `No Matrix defined!`

1.16 solve linear_system

Sintaxe: `solve linear_system;`

Este comando é utilizado para se resolver sistemas de equações lineares que são representados através de matrizes aumentadas.

Suponha que se deseje resolver o seguinte sistema de equações lineares:

$$\begin{cases} 10x_1 + 2x_2 + 1x_3 = 7 \\ 1x_1 + 5x_2 + 1x_3 = -8 \\ 2x_1 + 3x_2 + 10x_3 = 6 \end{cases}$$

Tal sistema de equações lineares pode ser representado pela seguinte equação matricial:

$$\begin{bmatrix} 10 & 2 & 1 \\ 1 & 5 & 1 \\ 2 & 3 & 10 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ -8 \\ 6 \end{bmatrix}$$

Tal equação matricial pode ser representada pela seguinte matriz aumentada:

$$\begin{bmatrix} 10 & 2 & 1 & 7 \\ 1 & 5 & 1 & -8 \\ 2 & 3 & 10 & 6 \end{bmatrix}$$

A matriz aumentada pode ser inserida no `dcmat` e o sistema linear resolvido através do comando `solve linear_system` utilizando, por exemplo, o método da fatoração LU. A resolução do sistema linear apresentado seria:

```
>matrix = [[10,2,1,7],[1,5,1,-8],[2,3,10,6]];
>solve linear_system;
```

Matrix x:

```
1.000000
-2.000000
1.000000
```

```
>
```

No exemplo apresentado a matriz x contém a solução do sistema linear, ou seja:

$$\begin{aligned} x_1 &= 1 \\ x_2 &= -2 \\ x_3 &= 1 \end{aligned}$$

Os sistemas lineares são classificados em 3 formas:

- **SPD** - Sistema Possível e Determinado
- **SPI** - Sistema Possível e Indeterminado
- **SI** - Sistema Impossível

Quando o sistema é possível e determinado, o sistema será resolvido e sua resposta mostrada, como no exemplo anterior. Se o sistema for possível e indeterminado, isto é, possui infinitas soluções, o programa deve mostrar a mensagem **SPI - The Linear System has infinitely many solutions**, como no exemplo a seguir:

```
>matrix=[[1,1,8],[2,2,16]];
>solve linear_system;
```

```
SPI - The Linear System has infinitely many solutions
```

```
>
```

Se o sistema for impossível, isto é, o sistema não possui nenhuma solução, o programa deve mostrar a mensagem **SI - The Linear System has no solution**, como no exemplo a seguir:

```
>matrix = [[3,2,6],[3,2,12]];
>solve linear_system;
```

```
SI - The Linear System has no solution
```

```
>
```

1.17 about

Sintaxe: `about;`

Este comando mostra informações sobre o desenvolvedor. Exemplo:

```
>about;
```

```
+-----+
|                                     |
|          DCMAT - CopyRight DC-UEL          |
|                                     |
+-----+
```

```
>
```

IMPORTANTE: Você deve mostrar o seu número de matrícula seguido do seu nome dentro da caixa de texto ao invés da mensagem aqui apresentada.

2 Mensagens de Erro

2.1 Caracteres Inválidos

Quando um caractere inválido for inserido, o sistema deve apresentar a mensagem **Simbolo Invalido** -> e apresentar o caractere inválido. Exemplo do erro:

```
>@  
Simbolo Invalido -> @  
  
>
```

2.2 Erros de Sintaxe

Quando houver um erro de sintaxe, o sistema deve apresentar a mensagem **Erro de sintaxe:** e apresentar entre colchetes o *token* que gerou o erro. Exemplo do erro:

```
>matrix = ;  
Erro de Sintaxe: [;]  
  
>
```

2.3 Tamanho Inválido de Matriz

O dcmat suporta matrizes com tamanho máximo de 10x10, isto é 10 linhas por 10 colunas. Ao se tentar inserir uma matriz fora destes limites, a mensagem **ERROR: Matrix limits out of boundaries.** é apresentada pelo sistema e nenhuma matriz é inserida. Exemplo do erro:

```
>matrix = [[0,1,2,3,4,5,6,7,8,9,0]];  
  
ERROR: Matrix limits out of boundaries.  
  
>
```

2.4 Ajuste da Área de plotagem

Ao se ajustar os valores que serão utilizados na visualização horizontal e vertical do gráfico, o primeiro valor deve ser menor que o segundo valor, caso contrário uma mensagem de erro deve ser exibida. Exemplo:

```
>set v_view 4:-1;  
  
ERROR: v_view_lo must be smaller than v_view_hi  
  
>  
>set h_view 1:-3.4;  
  
ERROR: h_view_lo must be smaller than h_view_hi  
  
>
```

2.5 Integral: Passo no cálculo da Integral

Caso o método que calcula a integral numericamente utilize passos e se deseje mudar os passos do algoritmo, o valor passado deve ser positivo, caso contrário a seguinte mensagem é mostrada:

```
>set integral_steps -2;

ERROR: integral_steps must be a positive non-zero integer

>
```

2.6 Integral: Limite inferior e superior

Ao se calcular uma integral numericamente, o limite inferior deve ser menor ou igual que o limite superior, caso contrário a seguinte mensagem é mostrada:

```
>integrate(1:-1,cos(x));

ERROR: lower limit must be smaller than upper limit

>
```

Se o limite inferior for igual ao limite superior a integral deve retornar o valor zero.

2.7 Determinante: Matriz não quadrada

O determinante só pode ser determinado para matrizes quadradas. Ao se tentar calcular o determinante de uma matriz não quadrada, a seguinte mensagem é mostrada:

```
>matrix = [[1,2]];
>solve determinant;

Matrix format incorrect!

>
```

2.8 Sistema Linear

Na resolução de um sistema linear de n variáveis, a matriz deve possuir n linhas e $n + 1$ colunas, caso contrário a seguinte mensagem é mostrada:

```
>matrix = [[1]];
>solve linear_system;

Matrix format incorrect!

>
```

3 Valores Padrão das Variáveis internas

Os valores padrão das variáveis internas são:

```
h_view_lo: -6.500000
h_view_hi: 6.500000
v_view_lo: -3.500000
v_view_hi: 3.500000
integral_steps: 1000
```

```
Draw Axis: ON.
Connect Dots: OFF.
```

Eles podem ser modificados livremente pelo usuário. Ao se digitar o comando `reset settings`, as variáveis internas tem o seu valor restabelecido aos valores aqui apresentados.

4 Critérios de Correção

Cada comando/mensagem de erro será testado 3 vezes, sendo a nota final proporcional a quantidade de comandos/mensagens que funcionaram corretamente. Os comandos serão inseridos continuamente dentro de uma mesma execução do programa, seguindo uma ordem aleatória.

5 Especificações de Entrega

O trabalho deve ser entregue no AVA em um arquivo `.zip` com o nome `dcmat.zip`. Este arquivo `.zip` deve conter somente os arquivos necessários à compilação, sendo que deve haver um `Makefile` para a geração do executável.

A entrega deve ser feita exclusivamente no AVA até a data/hora especificada. Não serão aceitas entregas atrasadas ou por outro meio que não seja o AVA.

Observação: o arquivo `.zip` não deve conter pastas, para que quando descompactado, os fontes do trabalho apareçam no mesmo diretório do `.zip`. O nome do executável gerado pelo `Makefile` deve ser `dcmat`.

IMPORTANTE: Arquivos ou programas entregues fora do padrão receberão nota ZERO. Entende-se como arquivo fora do padrão aquele que tenha um nome diferente de `dcmat.zip`, que contenha subpastas ou não seja um `.zip`, por exemplo. Entende-se como programa fora do padrão aquele que não contiver um `Makefile`, que apresentar erro de compilação ou o nome do executável for diferente de `dcmat`, por exemplo.