

Questão 1

Considere o somatório com n termos definido a seguir:

$$\frac{x+y}{1!} + \frac{3x-7y}{2!} + \frac{5x+13y}{3!} + \frac{7x-19y}{4!} + \frac{9x+25y}{5!} + \dots$$

Escreva um programa Scilab que solicite ao usuário o valor de n , calcule e imprima o valor desse somatório, considerando que $x = 50$ e $y = -80$. Seu programa deve verificar se o valor de n digitado pelo usuário é um número positivo, $n > 0$, solicitando repetidamente um novo valor, caso o valor digitado não seja válido. Você pode supor que o valor digitado pelo usuário será sempre um número inteiro (você apenas precisa testar se esse valor é positivo).

A seguir, dois exemplos de execução do programa.

Execução 1

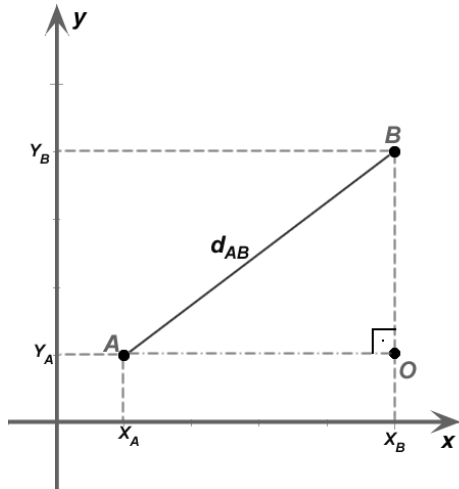
```
INFORME O NÚMERO DE PARCELAS: 0
VALOR INVÁLIDO PARA n !
INFORME O NÚMERO DE PARCELAS: -6
VALOR INVÁLIDO PARA n !
INFORME O NÚMERO DE PARCELAS: 1
VALOR DO SOMATÓRIO COM 1 PARCELAS: -30
```

Execução 2

```
INFORME O NÚMERO DE PARCELAS: 4
VALOR DO SOMATÓRIO COM 4 PARCELAS: 271.25
```

Questão 2

A distância entre dois pontos, A e B , no plano cartesiano pode ser dada pela fórmula:



$$distância = \sqrt{(X_B - X_A)^2 + (Y_B - Y_A)^2}$$

Escreva um programa principal e uma função definida pelo usuário, **DIST**, onde:

1. o programa principal faz a leitura das coordenadas dos pontos A e B , ou seja, os valores de X_A , Y_A , X_B e Y_B , conforme o exemplo de execução abaixo;
2. o programa principal faz a chamada a uma função **DIST**, com os valores das coordenadas lidas. A função retorna para o programa principal o valor numérico da distância entre os pontos A e B .
3. o programa principal faz a impressão da distância calculada.

OBS.: não é necessária a validação dos dados de entrada, as coordenadas serão sempre números reais.

A seguir, um exemplo de execução do programa.

Execução

CÁLCULO DA DISTÂNCIA ENTRE DOIS PONTOS

INFORME XA: 1

INFORME YA: 1

INFORME XB: 4

INFORME YB: 5

DISTÂNCIA ENTRE (1, 1) e (4, 5) : 5

Questão 3

Em uma tubulação projetada para escoamento de água, deve-se avaliar uma grandeza denominada “perda de carga”, a qual depende da vazão de líquido, do diâmetro e da rugosidade do tubo. Utiliza-se para isso a fórmula de Hazen-Williams:

$$J = Q^{1,85} \times 10,64 \times D^{-4,87} \times C^{-1,85}$$

onde:

J = perda de carga (m/m);
Q = vazão (m³/s);
D = diâmetro do tubo (m);
C = coeficiente de rugosidade.

Escreva um programa que leia a vazão desejada em um projeto de tubulação e a perda de carga máxima permitida. A seguir, o programa deve ler o diâmetro e o coeficiente de rugosidade de vários tubos, até encontrar um que proporcione perda de carga abaixo do limite permitido.

Não é necessário verificar a validade dos dados fornecidos pelo usuário.

A seguir, um exemplo de execução do programa.

Execução 1

```
Informe a vazão desejada (m3/s): 0.1
Informe a perda de carga máxima permitida (m/m): 0.4

Informe o diâmetro do tubo (m): 1
Informe o coeficiente de rugosidade do tubo: 0.1
O tubo não atende as especificações! Tente novamente.

Informe o diâmetro do tubo (m): 1.05
Informe o coeficiente de rugosidade do tubo: 0.05
O tubo não atende as especificações! Tente novamente.

Informe o diâmetro do tubo (m): 2
Informe o coeficiente de rugosidade do tubo: 0.1

O tubo com diâmetro 2 m e coeficiente de rugosidade 0.1
atende as especificações com perda de carga de 0.363853 m/m
```

Questão 4

Escreva um programa para ler diversos números inteiros positivos e, para cada um deles, determinar e imprimir seus divisores. O processamento deverá terminar imprimindo a frase “FIM DO PROGRAMA”, quando o número inteiro digitado for o número -1. Não é necessário verificar se o número digitado é inteiro e positivo.

A seguir, um exemplo de execução do programa.

Exemplo de execução

```
Digite um número inteiro positivo (-1 para terminar): 15
Divisores de 15:    1    3    5    15

Digite um número inteiro positivo (-1 para terminar): 36
Divisores de 36:    1    2    3    4    6    9    12    18    36

Digite um número inteiro positivo (-1 para terminar): -1

FIM DO PROGRAMA
```