

Tratamento de Exceções
------------------------

1. Crie uma classe `ExcecaoSoma` que receba um valor `x` e `n` números. Depois disso o programa deve somar esses `n` números enquanto a soma não for superior a `x`. O programa deve imprimir o valor somado (antes de atingir um número maior que `x`) e deverá informar quantos números foram somados e qual a média. O seu programa deve ser implementado utilizando as seguintes regras:
  - Você deve incluir tratamentos de exceção para lidar com a entrada de dados. Exemplo: o valor informado deve ser um número, o programa não deve permitir a entrada de números menores ou iguais a zero, o valor `x` lido da entrada não pode ser maior que 100;
  - Quando a soma for superior a `x`, o programa deverá gerar uma exceção (com o uso do `throws`) do tipo `ExcecaoAcimaDeX` que basicamente informa o usuário acerca da exceção. Para isso você deve explorar hierarquia de classes comuns (a partir da classe `Exception` ou derivadas).
2. Crie uma classe `PosicoesVetor` para preencher valores de um vetor de inteiros com `y` posições. O usuário irá informar os `y` valores a serem inseridos e suas respectivas posições no array. O programa deve tratar as exceções quando for informada uma posição inexistente do vetor e quando o valor informado não for um número.
3. Crie uma classe `ArquivoIP` que leia um arquivo texto ("`entrada.txt`") contendo uma lista de endereços IP e gere um outro arquivo ("`ips.txt`"), contendo um relatório dos endereços IP válidos e inválidos. Lembrando que um endereço IP possui o formato `x.y.z.w`, onde `x`, `y`, `z` e `w` são números no intervalo `[0, 255]`. O seu programa deve prover tratamento de exceções para falhas na abertura/leitura e fechamento do arquivo (`std::ios::exceptions`).

**Exemplo do arquivo de entrada:**

```
200.135.80.9
192.168.1.1
8.35.67.74
257.32.4.5
85.345.1.2
1.2.3.4
9.8.234.5
192.168.0.256
```

**Exemplo do arquivo gerado como saída:**

```
[Endereços válidos:]
200.135.80.9
192.168.1.1
8.35.67.74
1.2.3.4

[Endereços inválidos:]
257.32.4.5
85.345.1.2
9.8.234.5
192.168.0.256
```

**Considerações!**

- Todos os exercícios devem conter `.h`, `.cpp`, e `main.cpp`; e o seu `main.cpp` deve conter, minimamente, instruções para criação (instanciação de objetos) e chamadas das funções implementadas.